

41<sup>1</sup>/<sub>2</sub>

# FUN PROJECTS FOR YOUR ATARI®

BY DALE DISHAROON & HERBERT KOHL



A CREATIVE PASTIMES BOOK

# **41<sup>1</sup>/<sub>2</sub>** **Fun Projects** **for the ATARI®**

**DALE DISHAROON**  
**and**  
**HERBERT KOHL**

Herbert Kohl, Series Adviser



A Creative Pastimes Book  
Reston Publishing Company  
A Prentice-Hall Company  
Reston, Virginia

**Library of Congress Cataloging in Publication Data**

Disharoon, Dale.

41 1/2 fun projects for the Atari home computer.

"A Creative Pastimes book."

I. Computer games. 2. Atari computer—Programming.  
3. Basic (Computer program language) I. Kohl, Herbert R.  
II. Title. III. Title: Forty-one and one-half fun  
projects for the Atari home computer.

GV1469.2.D576 1984

794.8'2

84-3329

ISBN 0-8359-2101-8

Interior design and production by Laura Cleveland.

This book is published by Reston Publishing Company which is not affiliated with Atari, Inc. and Atari is not responsible for any inaccuracies. ATARI is a registered trademark of Atari, Inc.

Copyright © 1984 by Reston Publishing Company, Inc.,  
Reston, Virginia 22090.

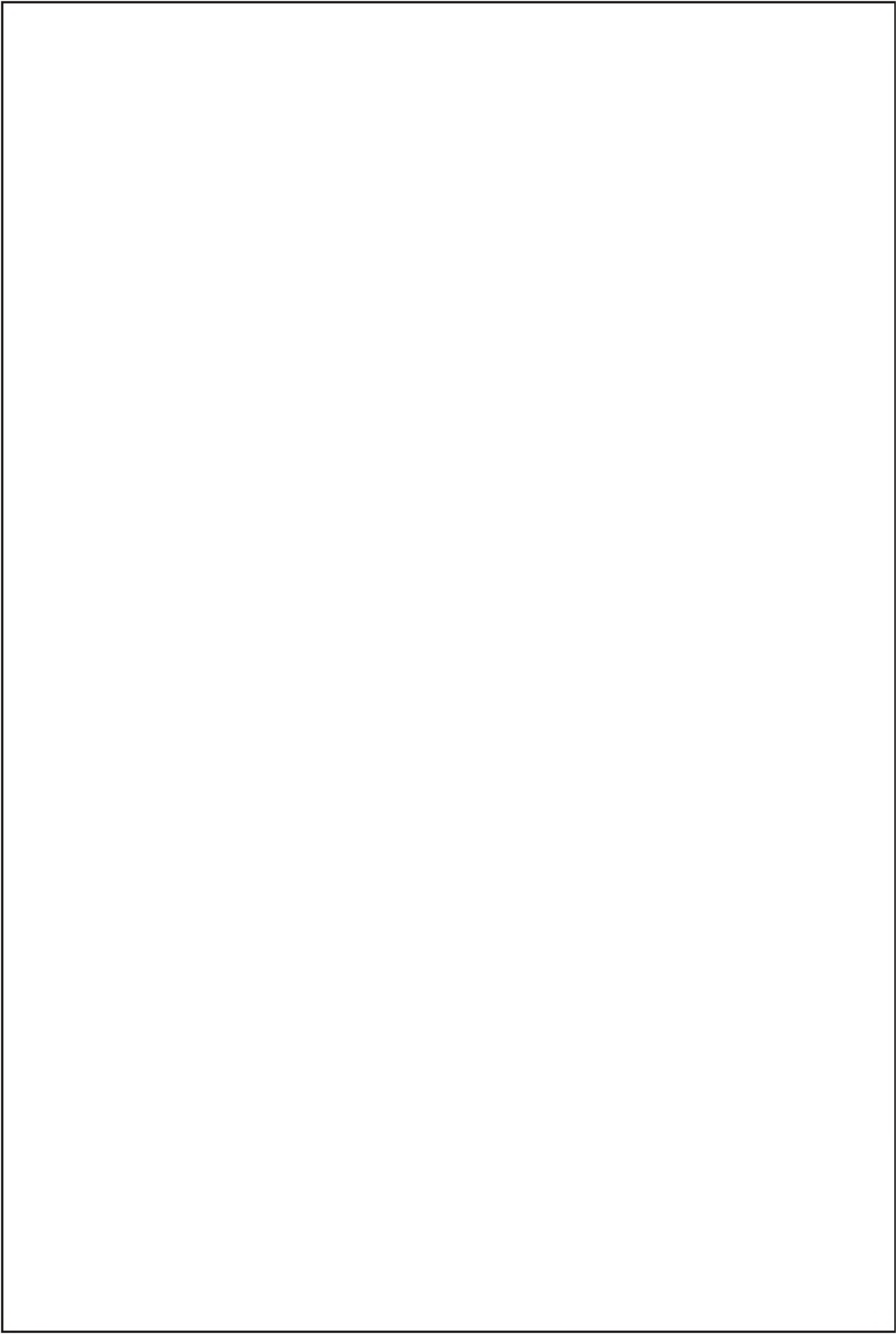
All rights reserved. No part of this book may be reproduced in any way or by any means without written permission from the publisher.

10 9 8 7 6 5 4 3 2 1

Printed in the United States of America.

# Contents

<b>Introduction</b>	<b>1</b>
<b>PART I: Word Play</b>	<b>3</b>
<b>PART II: Number Play</b>	<b>17</b>
<b>PART III: IQ Building</b>	<b>33</b>
<b>PART IV: Strategy Puzzles</b>	<b>53</b>
<b>PART V: Music and Noise</b>	<b>73</b>
<b>PART VI: Computer Utilities</b>	<b>91</b>
<b>1/2 of Many Things You Can Do with Your Computer</b>	<b>119</b>



# Introduction

Almost anything new and different can be exciting for a time. And almost all excitement wears off after a while. For many people, owning their own microcomputer is such an experience. At first there seem to be thousands of things to do to make life easier and more interesting. For many people, however, the magic fades and the computer becomes simply a checkbook balancer and game player if it doesn't actually end up in the closet. The intent of this book is to provide a new infusion of excitement about what you can do with your home computer.

With one exception, the programs in this book are quite short, easy to copy, and easy to modify. The text will suggest ways you can tailor the programs for your own use—for party games, data analysis, school work, or learning to program. The 41 1/2 things to do range all the way from a dating program that will match you with a potentially congenial date to a mini-word processor, a Dungeons and Dragons dice rolling program, an excuse generator, and strategy and music games. It includes practical programs such as a guitar tuner and chord teacher, a metric converter, and a comparison shopping program. It also contains party games and other activities that you can do with your friends. Underlying all of these programs are the ideas that you can modify a program and change it to suit your purposes and that programming can be fun and lead to fun as well.

The central part of the book provides you with a number of simple computer utilities. The most common utilities in our lives are gas, electricity, and the telephone. They are utilities in that they can be used in many different ways. They provide the energy or the instrument and you determine the particular use. The same thing is true of computer utilities. They provide a form that you can use in many different ways with a simple input of your own. Thus, the balloting program in this book can be used for club elections, picking the top ten records, doing personal preference polls, etc. The utility gives you the structure and you provide the content. The same is true of the graph generating program. It can be used to measure sales and inventories, to compare school grades, to illustrate the results of

science experiments, or for opinion surveys. The text will suggest some applications for the utilities and show you how to list and change the programs in the book. You should feel free to go beyond the text and create your own modifications, and, maybe, build your own utilities.

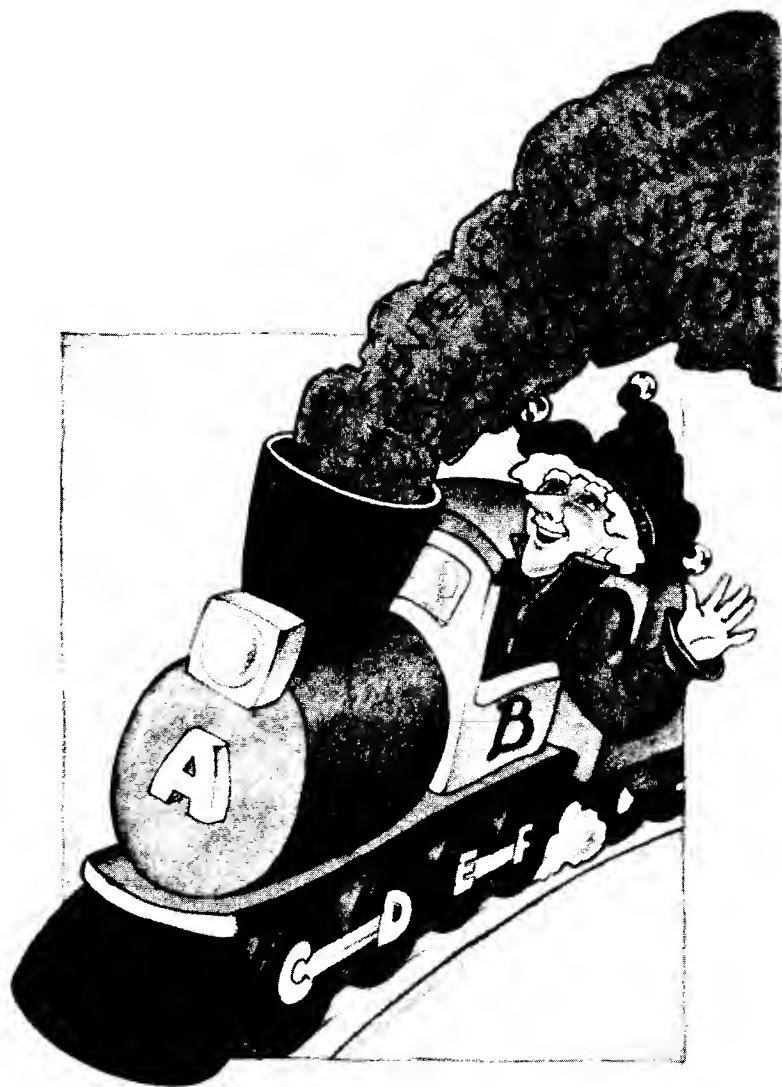
All of the programs in this book fit on one disk. With the exception of the program for the Dozo game, they are easy to copy and take up less than two pages of text. It would make sense for you to copy and save on disk those programs you want to use or modify. Copying the same program more than once is a bore and you can always save your original copy as well as your modifications. After you copy a program, however, run it several times to make sure that some simple copying error didn't slip in. Try all the options the programs offer. Play to win, to deliberately lose, and try any way you can to trick the program into bombing. We've done this ourselves and are confident that the programs work. But, human error always has a way of creeping in. We have been careful to eliminate ours and believe it makes sense for you to take the extra time to check your work.

Finally, for the 1/2 of a thing you can do with your computer. Actually it is not one-half of a thing but many halves—many ideas that you can turn into programs as you become comfortable with BASIC. These final ideas are just starters, teasers, suggestions of ways in which you can focus the use of your programming skills on creating programs others will enjoy using.

# PART I

---

## Word Play





**T**he programs in this section all fool around with words. They range from an anagram generator to an excuse generator and include word hunt puzzles, a sentence scrambler, and even an arcade-type game that will teach you how to use the keyboard. Many of the programs are easily modified and we encourage you to copy them and change them in any way you like.

# 1

## Joystick Word Hunt

You probably have seen many word hunt puzzles. They consist of a square or rectangle of letters in which a number of words are hidden either vertically, horizontally, or diagonally, and backwards and forwards. Here is a very simple word hunt grid and the list of words hidden in it:

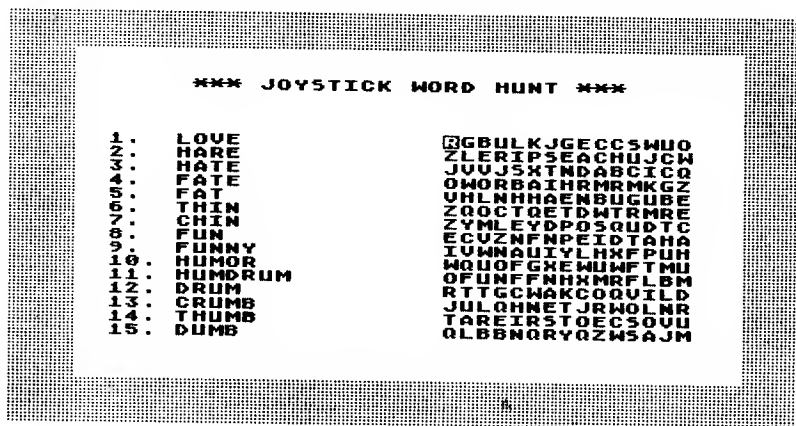
hidden words: last, cat, rat  
curl, pot, coat

hidden word grid:

c o r l  
o a a r  
a s t u  
t o p c

The object of the word hunt is to find the words on the list hidden in the grid.

The program for Joystick Word Hunt allows you to choose any 15 words of up to 15 letters long. Then, the computer automatically generates a word grid for you or your friend to solve. Here is a sample list and the grid the computer generated. You'll find that even if you chose the words yourself it is not easy to find them all in the grid.



You need a joystick in port 1 to play this game. Move the flashing cursor on the game board to the first letter of any word you find. Press the joystick button. A message will appear: WORD NUMBER. Type only the number that corresponds to the word you've found. Press **RETURN**. If you've found the correct match, the word will appear in inverse characters on the grid. Keep playing until you find all 15 original words.

```

100 REM *** JOYSTICK WORD HUNT ***
110 GOTO 700
200 GOSUB 900:FOR M=0 TO 14:L=L(M):T$=
W$(M*15+1,M*15+L):POSITION 2,6+M: ? M+1
;". "
210 XD=INT(RND(1)*3-1):YD=INT(RND(1)*3
-1):IF XD=0 AND YD=0 THEN 210
220 X=INT(RND(1)*(15-(ABS(XD)*(L-1))))
:IF XD<0 THEN X=14-X
230 Y=INT(RND(1)*(15-(ABS(YD)*(L-1))))
:IF YD<0 THEN Y=14-Y
240 XP=X:YP=Y:FOR T=1 TO L:IF XP<0 OR
XP>14 OR YP<0 OR YP>14 THEN 210
250 POSITION 22+XP,6+YP:GET #2,K:IF K<
>32 AND K<>ASC(T$(T,T)) THEN 210
260 XP=XP+XD:YP=YP+YD:NEXT T:X(M)=X:Y(
M)=Y:XD(M)=XD:YD(M)=YD:FOR T=1 TO L:PO
SITION 22+X,6+Y:PUT #2,ASC(T$(T,T))
270 X=X+XD:Y=Y+YD:NEXT T:POSITION 6,6+
M: ? T$:NEXT M
300 FOR X=0 TO 14:FOR Y=0 TO 14:POSITI
ON 22+X,6+Y:GET #2,K:IF K=32 THEN POSI
TION 22+X,6+Y:PUT #2,INT(RND(1)*26)+65
310 NEXT Y:NEXT X:X=0:Y=0:WW=0
400 Z=STICK(0):IF STRIG(0)=0 THEN 500
410 IF Z>8 AND Z<12 THEN X=X-1:IF X<0
THEN X=14
420 IF Z>4 AND Z<8 THEN X=X+1:IF X>14
THEN X=0
430 IF Z=10 OR Z=14 OR Z=6 THEN Y=Y-1:
IF Y<0 THEN Y=14
440 IF Z=9 OR Z=13 OR Z=5 THEN Y=Y+1:I
F Y>14 THEN Y=0
450 IF Z<>15 THEN POKE 53279,0
460 POSITION 22+X,6+Y:GET #2,K:POSITIO
N 22+X,6+Y:IF K<128 THEN PUT #2,K+128
470 IF K>127 THEN PUT #2,K-128
480 FOR T=1 TO 4:NEXT T:POSITION 22+X,
6+Y:PUT #2,K:GOTO 400
500 TRAP 520:POKE 764,255:POSITION 4,4
: ? "WORD NUMBER";:INPUT G:G=G-1

```

```

510 IF X(G)=X AND Y(G)=Y THEN 600
520 GOSUB 910:GOTO 400
600 XP=X:YP=Y:T#=W$(G*15+1,G*15+L(G)):
FOR T=1 TO L(G):POSITION 22+XP,6+YP:PU
T #2,ASC(T$(T,T))+128
610 POSITION 5+T,6+G:PUT #2,ASC(T$(T,T
))+128:XP=XP+XD(G):YP=YP+YD(G):NEXT T:
GOSUB 910
620 X(G)=-1:WW=WW+1:IF WW<15 THEN 400
630 POSITION 4,4:? "YOU DID IT! PRESS
START"
640 IF PEEK(53279)<>6 THEN 640
650 RUN
700 OPEN #2,4,0,"S:":DIM W$(225),T$(15
),L(14),X(14),Y(14),XD(14),YD(14):GRAP
HICS 0:POKE 752,1:OPEN #1,4,0,"K:"
800 GOSUB 900:POSITION 2,5:? "ENTER FI
FTEEN WORDS..."
810 FOR T=0 TO 14:POSITION 2,7+T:? T+1
;".":POSITION 6,7+T:INPUT T$:IF T$=""
THEN T=T-1:NEXT T
820 L(T)=LEN(T$):W$(T*15+1,T*15+L(T))=
T$:NEXT T:POSITION 2,5:? "ARE YOUR WOR
DS CORRECT?"
830 GET #1,K:IF CHR$(K)="Y" THEN 200
840 GOTO 800
900 ? CHR$(125):POSITION 7,2:? "*** JO
YSTICK WORD HUNT ***":RETURN
910 POSITION 4,4:? " "
RETURN

```

## 2

### Anagrams

An anagram is a scramble of letters. For example, the word *and* has the following anagrams:

nad  
nda  
adn  
dna  
dan

Sometimes the anagram of a word gives rise to other words. For example, consider these two sets of anagrams:

word: top  
anagrams: pot  
pta  
tpa  
opt  
otp

In this case, two anagrams of "top" are also words—"opt" and "pot."

word:	meat	
anagrams:	team	eatm
	tema	eamt
	tmea	emat
	tmae	emta
	tame	etma
	taem	etam
	aemt	meta
	aetm	mate
	amet	maet
	amte	mtae
	atem	mtea
	atme	

Of the 24 variations, only three and at most four (if you accept *meta* as a word) make up words. An interesting and, so far as we know, unsolved word problem is whether there is a four letter word all of whose anagrams make up other words. You could easily write a computer program that could generate all the anagrams of any given word using string variables. However, the program here is more of a games form. You type in any word and the computer scrambles it. Then ask a friend to try to unscramble it. The computer will keep score and tell your friend when he or she has the correct unscrambling.

```
100 REM          *** ANAGRAMS ***
110 DIM W$(30),A$(30),B$(30)
120 GRAPHICS 0:POKE 752,1
130 PRINT CHR$(125):POSITION 12,3:PRIN
T "*** ANAGRAMS ***"
```

```

140 POSITION 2,9:PRINT "ENTER YOUR WOR
D..."
150 POSITION 2,11:PRINT CHR$(156):POSIT
ION 2,11:INPUT W$:IF W$="" THEN 150
160 POSITION 2,9:PRINT "TRY TO SOLVE T
HIS ANAGRAM...":POSITION 2,11:PRINT CH
R$(156)
200 B$(1)=" ":B$(30)=" ":B$(2)=B$:L=LE
N(W$):A$="":FOR T=1 TO L
210 R=INT(RND(1)*L)+1:IF B$(R,R)="*" T
HEN 210
220 B$(R,R)="*":A$(T,T)=W$(R,R):NEXT T
:IF A$=W$ AND L>1 THEN 200
230 POSITION 2,11:PRINT A$:B=20-LEN(W$
)/2:POSITION B,6:FOR T=1 TO L:PRINT CH
R$(95);:NEXT T
240 TRAP 240:POSITION 2,14:PRINT CHR$(
156):POSITION 2,14:INPUT B$
250 FOR T=1 TO L:POSITION B+T-1,6:IF B
$(T,T)=W$(T,T) THEN PRINT W$(T,T)
260 NEXT T:IF B$<>W$ THEN 240
270 POSITION 15,18:PRINT "THAT'S IT!"
300 POKE 764,255
310 IF PEEK(764)=255 THEN 310
320 POKE 764,255:GOTO 130

```

# 3

---

## Sentence Scrambler

This program will take a sentence and scramble all of the words in it. For short sentences, the unscrambling is quite easy. For example:

walk. took a l

quite easily unscrambles into:

I took a walk.

Try this one, however:

at time the You stars. tell looking by the can

When you type in the program be very careful about capitals, punctuation, and spacing. The computer responds to all of these different aspects of writing. Spaces between words are particularly crucial in mixing up sentences because sensing spaces is the only way the computer can tell where a word ends. The computer cannot read, at least yet. A blank space is just another symbol to it, one which, in this program, indicates where a unit to be scrambled ends.

The main functions used in this program are LEN(A\$) and A\$(X,X). LEN(A\$) tells you the length of any word you store in the computer's memory. For example, in Atari BASIC, if you dimension A\$ and input the word "FINE" as in the following program:

```
10 DIM A$(20)
20 INPUT A$
```

then      LEN(A\$)=4

The function A\$(X,X) lets you print out any letter or sequence of letters in A\$. Thus, using the same example:

```
        A$(1,1)=F
and      A$(2,3)=IN
```

Here is a simple educational program illustrating the use of LEN and A\$:

```
10 DIM A$(26),B$(1)
20 A$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
30 PRINT "HOW MANY LETTERS ARE IN THE ALPHABET?"
40 INPUT X
50 IF X= LEN (A$) THEN PRINT "GOOD. NOW TRY THIS:"
60 LET Y=INT(RND(1)*26)+1
70 PRINT "WHAT LETTER COMES ";Y;"th in the alphabet?"
80 INPUT B$
90 IF B$=A$(Y,Y) THEN PRINT "RIGHT."
100 IF B$<> A$(Y,Y) THEN PRINT "I'M SORRY YOU MISSED."
```

This program checks the length of A\$ and matches it with the number you put in for X. Then it picks a random position out of A\$ using the Y variable and asks you which letter corresponds to that

position. This is just a simple use of LEN and A\$(X,X). Play with these functions and look at your BASIC Reference Manual for more information on how to use them.

Now here is the scrambled sentence program:

```
100 REM *** SENTENCE SCRAMBLER ***
110 DIM L(29),M(29),A$(600),S$(120),B$(120):GRAPHICS 0:POKE 752,1
120 PRINT CHR$(125):POSITION 7,3:PRINT
   "*** SENTENCE SCRAMBLER ***"
130 POSITION 2,7:PRINT "ENTER YOUR SENTENCE...":PRINT :PRINT
140 POKE 752,0:INPUT S$:POKE 752,1:GOSUB 500
150 A$(1)=" ":A$(600)=" ":A$(2)=A$:B$=A$:L=LEN(S$):C=0:P=1:EE=0
160 FOR T=1 TO L:IF S$(T,T)=" " THEN 200
170 IF S$(T,T)="." OR S$(T,T)="!" OR S$(T,T)="?" THEN EE=1:GOTO 200
180 NEXT T:T=T-1:EE=1
200 L(C)=T-P+1:A$(C*20+1,C*20+20)=S$(P,T):IF EE=0 THEN C=C+1:P=T+1:NEXT T
300 L(C)=L(C)+1:FOR T=0 TO C
310 R=INT(RND(1)*(C+1))+1:IF B$(R,R)="*" THEN 310
320 B$(R,R)="*":M(R-1)=T:NEXT T
330 POSITION 2,7:PRINT "TRY TO UNSCRAMBLE THIS...":PRINT :PRINT
340 FOR T=0 TO C:PRINT A$(M(T)*20+1,M(T)*20+L(M(T))):NEXT T
350 POSITION 2,14:POKE 752,0:INPUT B$:POKE 752,1
360 IF B$<>S$ THEN GOSUB 600:POSITION 15,21:PRINT "TRY AGAIN":GOTO 350
370 POSITION 14,21:PRINT "YOU DID IT!":FOR J=17 TO 9 STEP -1:FOR T=14 TO 0 STEP -2:SOUND 0,J,10,T:NEXT T:NEXT J
400 POKE 764,255
410 IF PEEK(764)=255 THEN 410
420 POKE 764,255:GOTO 120
500 FOR T=1 TO 8:POSITION 2,7:PRINT CHR$(156):NEXT T:RETURN
600 FOR T=1 TO 8:POSITION 2,14:PRINT CHR$(156):NEXT T:RETURN
```



# 4

## Alphawiz

This is the basic shoot 'em up arcade game in its simplest form. The game displays the alphabet at the bottom of the screen. A target appears above one of the letters at the top of the screen. You have to press the correct letter to hit the target. The letters have a small amount of fire power and you will see your hits and misses on the screen. You get five points per hit and lose five per miss.

If you want to play a little code game, take a look at the PRINT statement at line 20. It tells the computer to print out the alphabet in order at the bottom of the screen. Now, if you changed that line using a simple shift code as follows:

ABCDEFGHIJKLMNOPQRSTUVWXYZ  
ZABCDEFGHIJKLMNOPQRSTUVWXYZ

then, if you pressed A the Z would shoot; if you pressed B the A would shoot, etc. Your shooting would then be in code. After a while you could probably figure out the structure of the code.

```
100 REM          *** ALPHAWIZ ***
110 DIM TX(10),TY(10):OPEN #1,4,0,"K:"
120 GRAPHICS 5:SETCOLOR 1,9,8:SETCOLOR
    2,0,0:POKE 752,1:PRINT "      ABCDEFGH
    IJKLMNOPQRSTUVWXYZ"
130 COLOR 1:PLOT 1,39:DRAWTO 1,0:DRAWTO
    0 78,0:DRAWTO 78,39:POKE 20,0:POKE 19,
    0:POKE 764,255
140 TX=INT(RND(1)*26)*2+14:FOR T=0 TO
    10:TX(T)=RND(1)*8-4+TX:TY(T)=RND(1)*8+
    7:NEXT T
150 POSITION TX,11:PRINT #6;"BB":POSIT
    ION TX,10:PRINT #6;"BB"
160 POKE 656,2:POKE 657,4:PRINT "SCORE
    : ";S;" "
170 IF PEEK(764)<>255 THEN 200
180 IF PEEK(19)>14 THEN 400
190 GOTO 170
200 GET #1,K:IF K<65 OR K>90 THEN POKE
    702,64:POKE 694,0:GOTO 170
```

```

210 K=2*K-116:FOR Y=39 TO 1 STEP -2:CO
LOR 1:PLOT K,Y:SOUND 0,100-2*Y,8,Y/3
220 IF Y=11 THEN LOCATE TX,Y,A:IF A=1
THEN 300
230 COLOR 0:PLOT K,Y:NEXT Y:SOUND 0,0,
0,0:S=S-5:GOTO 160
300 COLOR 2:SETCOLOR 1,4,8:FOR K=50 TO
150 STEP 5:V=15-K/10:SOUND 0,K,8,V:SO
UND 1,K+5,8,V
310 PLOT TX(K/10-5),TY(K/10-5):SETCOLO
R 4,0,V:SETCOLOR 2,0,V:NEXT K
320 COLOR 0:FOR T=0 TO 10:PLOT TX(T),T
Y(T):NEXT T:POSITION TX,11:PRINT #6;"
":POSITION TX,10:PRINT #6;" "
330 SETCOLOR 1,9,8:S=S+5:GOTO 140
400 POKE 656,2:POKE 657,17:PRINT "***
GAME OVER ***";:IF S>HS THEN HS=S
410 POKE 656,3:POKE 657,4:PRINT "**HIGH
":HS;:POKE 656,3:POKE 657,21:PRINT "
Press START";CHR$(253);:S=0
420 IF PEEK(53279)<>6 THEN 420
430 GOTO 120

```

# 5

---

## Creative Name Machine

Did you ever wonder how people come up with names for computer games or businesses or products? Most of the process is free association and fiddling around with words. The computer can often do this as well as people, especially if you make your original lists of words interesting enough. This program generates some names you might like to use one day. Some of them are COSMOS WEAVER, CHASM SPINNER, and VAULT PRODUCER. If you look at lines 310 and 410 of the program you'll see the DATA statements that provide the words for the computer to play with. You can change these with any lists you have so long as you remember to separate the words by a comma. Line 140 reads a word from the list at 310 and line 150 reads a word from 410. The combination is what you see printed out. The



READ ... DATA .... statement in BASIC allows you to play around with lists of data.

Now try our names and add some twists of your own.

```
100 REM *** CREATIVE NAME MACHINE ***
110 DIM A$(20),B$(20)
120 GROUP1=12
130 GROUP2=10
140 RESTORE 300:FOR T=0 TO INT(RND(1)*
GROUP1):READ A$:NEXT T
150 RESTORE 400:FOR T=0 TO INT(RND(1)*
GROUP2):READ B$:NEXT T
160 POKE 752,1:PRINT CHR$(125):SETCOLO
R 2,INT(RND(1)*16),2
170 T=LEN(A$):T=T+LEN(B$)
180 POSITION 20-T/2,10:?" A$;" ";B$
190 POKE 764,255
200 IF PEEK(764)=255 THEN 200
210 GOTO 140
300 REM *** WORD GROUP 1 ***
310 DATA DUNGEON,VAULT,CRYPT,CAVERN,AD
VENTURE,WORLD,ENCOUNTER,SPACE,CHASM,UN
IVERSE,MICROCOSM,COSMOS
400 REM *** WORD GROUP 2 ***
410 DATA MACHINE,WEAVER,SPINNER,VENTUR
E,BUILDER,SYNTHESIZER,INVENTOR,CREATOR
,COMPOSER,PRODUCER
```

# 6

## Excuse Generator

Did you ever wish you had an automatic excuse generator that would cover you when you were late for work, forgot to do homework, or wanted to get out of a party or meeting? Well, here's an attempt at one. Basically, this program consists of an empty letter form in which you fill in the particulars. If you don't like the way we've done it, you change all of the PRINT statements in the program or add some new ones of your own. The program is a phoney personal letter generator. You can use it to create personalized business letters, invitations to parties, etc. Our excuse for including it in the book was to show what a letter or response form is like and how easily it can be changed on a computer.

```
DEAR PALLID,  
I'M VERY SORRY THAT I DON'T HAVE  
THE TIME. YOU SEE, MY  
BEST FRIEND LOST IT  
WHILE I PICKED FLEAS FROM HIS COAT.  
GEE, I SURE AM SORRY. I WON'T LET  
IT HAPPEN AGAIN. I PROMISE.  
  
SINCERELY,  
LIVID
```

```
100 REM *** EXCUSE GENERATOR ***  
110 DIM A$(50),B$(50),C$(50),D$(50),E$(  
50),F$(50)  
120 POKE 752,1:PRINT CHR$(125):POSITIO  
N 12,3:PRINT "EXCUSE GENERATOR"  
130 POSITION 2,6:PRINT "YOUR NAME: ";:  
INPUT A$:IF A$="" THEN A$="ME"  
140 PRINT:PRINT "PERSON TO GET EXCUSE  
: ";:INPUT B$:IF B$="" THEN B$="TEACHE  
R"
```

```

150 PRINT :PRINT "ITEM TO BE EXCUSED:
";:INPUT C$:IF C$="" THEN C$="HOMEWORK
"
160 PRINT CHR$(125):POSITION 2,3:PRINT
"DEAR ";B$;",";:PRINT :PRINT "I'M VERY
SORRY THAT I DON'T HAVE":PRINT
170 PRINT "THE ";C$;". YOU SEE, MY":P
RINT
180 RESTORE 300:FOR T=0 TO INT(RND(1)*
7):READ D$:NEXT T:RESTORE 400:FOR T=0
TO INT(RND(1)*7):READ E$:NEXT T
190 RESTORE 500:FOR T=0 TO INT(RND(1)*
7):READ F$:NEXT T
200 PRINT D$;" ";E$:PRINT :PRINT "WHIL
E I ";F$;".":PRINT :PRINT "GEE, I SURE
AM SORRY. I WON'T LET"
210 PRINT :PRINT "IT HAPPEN AGAIN. I
PROMISE.":PRINT :PRINT :PRINT "SINCERE
LY,":PRINT :PRINT A$
220 POKE 764,255
230 IF PEEK(764)=255 THEN 230
240 POKE 764,255:GOTO 120
300 DATA DOG,DAD,BABY BROTHER,OLDER BR
OTHER,CAT,BEST FRIEND,BOSS
400 DATA ATE IT,USED IT FOR KINDLING,S
LOBBERED ALL OVER IT,SHINED HIS SHOES
WITH IT,TORE IT TO PIECES
410 DATA STOLE IT,LOST IT
500 DATA PICKED FLEAS FROM HIS COAT,WA
S ON A TRIP,CHANGED HIS DIAPERS,WASN'T
WATCHING
510 DATA WAS FEEDING HIM CAT FOOD,STOO
D BY HELPLESSLY,WAS ASLEEP

```

# PART II

---

## Number Play



**T**his section contains several number games you may not have seen before as well as some conversion and sorting programs you can use for many different purposes. However, the first program in the section is a challenge that comes from Marlyn Burns, the author of the *I Hate Math Book* and *Math for Smarty Pants*. We translated her challenge into computer form. It can be fun to do similar things by taking books on mathematical games and recreations and giving those challenges a computer twist. Often, the computer can make the game more powerful because it can check challenges quickly, provide hints, tally scores, and set up timing devices.

# 7

---

## The \$1.00 Word Challenge

Here is the challenge in Marlyn Burns' own words:

"Excellent" is a \$1.00 word. Mathematically speaking, that is. So are "friendlier" and "grumpy" and "elephants" And so is "discipline," which is what it takes to find more of these \$1.00 wonders. To find out how much a word is worth, you give each letter a cent value: a=\$.01, b=\$.02, c=\$.03, and so on up to z=\$.26. Then you add the value of each letter in a word.

Check for yourself to make sure the words given above really are worth exactly \$1.00. Then you're on your own. How about starting with your first name? Is it worth \$1.00? What about your friends' names? Or the town or city you live in?

The \$1.00 word fever has been spreading throughout the United States. To date over 500 \$1.00 words have been found by addition-loving kids across the country. Some have been writing \$1.00 sentences—sentences in which every word is worth \$1.00. Some examples: "Prevent inflation." "Whenever Henrietta whistled, thirty trembling costumed elephants merrily performed."

Of course, dedicated \$1.00 word zealots have brought their computers into the search, programming them to calculate the value of whatever word they input. Try it. It takes all the drudge out of the search.

Some hints may be useful for getting you started. There is one Halloween word that is worth \$1.00 (It's not Halloween; that's only worth \$.95.) There's one Thanksgiving \$1.00 word, as well as one astrological sign, and one United States coin. There are at least two United States cities—one of them in Wisconsin, and the other in both Oregon and Maine. To date, one five-letter \$1.00 word has been found. No one, however, has submitted a word with four letters. (Is it possible to have a four-letter word worth \$1.00?)

There's a beverage that is illegal for minors to drink that's worth \$1.00, though it sometimes costs more than that to buy in restaurants and bars. There are several zoo animals that are \$1.00 words, and at



least one underwater creature. Something that rarely strikes twice in the same place, so they say, is also a \$1.00 word. And there is a number that is less than 100 that is worth \$1.00 when written as a word.

This program will check to see if your words are \$1.00 words. If your word doesn't quite make a dollar, the program will also tell you how much it is worth using the coding system where A is a penny, B two cents, etc.

```
100 REM *** ONE-DOLLAR WORDS ***
110 DIM A$(20):GRAPHICS 0:POKE 752,1
120 PRINT CHR$(125):POSITION 7,3:PRINT
    "THE ONE-DOLLAR WORD MACHINE"
130 PRINT :PRINT :PRINT "PLEASE ENTER
YOUR WORD...";
140 INPUT A$:IF A$="" THEN 130
150 V=0:FOR T=1 TO LEN(A$):IF A$(T,T)<
"A" OR A$(T,T)>"Z" THEN 130
160 V=V+ASC(A$(T,T))-64:NEXT T:PRINT :
IF V=100 THEN PRINT A$;" IS A ONE DOLL
AR WORD!":GOTO 130
170 PRINT A$;" IS WORTH ";V;" CENTS.":
GOTO 130
```

If you want to make a \$2.00 word machine or a half-dollar word machine, all you have to do is change the IF-THEN and PRINT commands on line 160. Thus, the following changes in line 160 will give a Fifty Cents Word Machine:

```
160 V=V+ASC(A$(T,T))-64:NEXT T:PRINT:IF V=50 THEN PRINT
A$;" IS A FIFTY CENT WORD!":GOTO 130
```

# 8

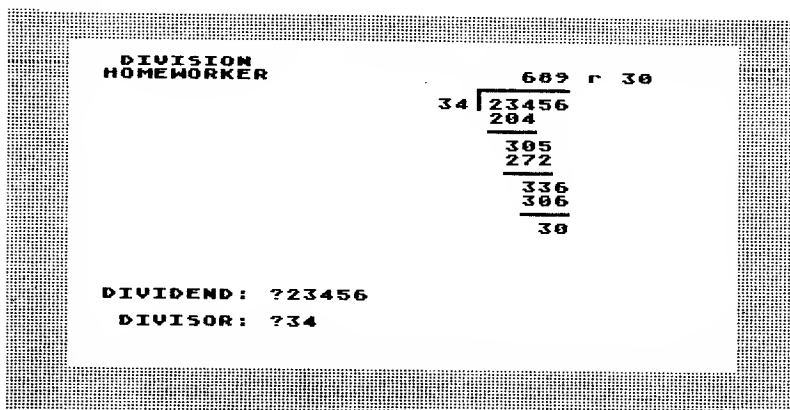
---

## Division Homeworker

Did you ever get frustrated when you were asked to "show the work" when you were doing math homework? After all, a calculator or com-

puter could do the work. Why bother with the tediousness of long division anymore? There is even a rumor that long division is a device created by teachers to torture their students with boredom. We've come up with a solution to the long division nightmare. Here is a program that not merely gives you the answer to a long division problem, remainder and all, but also "shows the work." We hope it will save you hours of boredom that could be used in more productive and pleasurable activity.

The program has another value. It will take you step by step through long division problems and can be a useful way to learn the process.



```

100 REM *** DIVISION HOMEWORKER ***
110 DIM N$(9),A$(9),T$(9):X=24:Y=4
120 TRAP 120:POKE 752,1:PRINT CHR$(125)
:PRINT " DIVISION":PRINT "HOMEWORKER"
:POSITION 2,18:PRINT "DIVIDEND: ";
130 INPUT N:A$=STR$(N):POSITION 3,20:P
RINT "DIVISOR: ";:INPUT D:IF D>N OR D=
0 OR N>999999 THEN 120
140 POSITION X,Y:PRINT CHR$(124);CHR$(
30);CHR$(28);CHR$(17);
150 FOR T=1 TO LEN(A$):PRINT CHR$(18);
:NEXT T:POSITION X-LEN(STR$(D)),Y:PRIN
T D:POSITION X+1,Y:PRINT N:GOSUB 500
160 NS=1:QP=0:BP=0:FF=0:N$=A$(1,1)
170 NT=VAL(N$(1,NS)):IF NT>=D THEN 210
180 QP=QP+1:IF FF=1 THEN POSITION X+QP
,Y-2:PRINT "0":GOSUB 500
190 NS=NS+1:IF NS>LEN(A$) THEN R=NT:GO
TO 600

```

```

200 N$=A$(1,NS):POSITION 2+X+QP-LEN(N$
),Y+BP:PRINT N$:GOTO 170
210 FF=1:QT=INT(NT/D):R=NT-D*QT:QP=QP+
1:POSITION X+QP,Y-2:PRINT QT:GOSUB 500
220 T=1+X+QP-LEN(STR$(D*QT)):POSITION
T,Y+BP+1:PRINT D*QT:GOSUB 500
230 POSITION T,Y+BP+2:FOR T=1 TO LEN(S
TR$(D*QT)):PRINT CHR$(18);:NEXT T
240 POSITION 1+X+QP-LEN(STR$(R)),Y+BP+
3:PRINT R:GOSUB 500
250 IF NS=LEN(A$) THEN 600
260 T$=STR$(R):T$(LEN(T$)+1)=A$(NS+1,N
S+1):POSITION 1+X+QP-LEN(STR$(R)),Y+BP
+3:PRINT T$:N$=T$:GOSUB 500
270 T$=STR$(R):T$(LEN(T$)+1)=A$(NS+1,L
EN(A$)):A$=T$:NS=LEN(N$):BP=BP+3:GOTO
170
500 POKE 764,255:FOR K=1 TO 50:NEXT K
510 IF PEEK(764)=255 THEN 510
520 POKE 764,255:RETURN
600 POSITION X+QP+2,Y-2:PRINT "r ";R:G
OSUB 500:GOTO 120

```

# 9

## Kabala Name Game

According to the Kabala, a mystical Jewish treatise about the relationship between numbers, letters, and things spiritual, every word has a numerical value which determines its sacredness and quality. This is especially true for names. We've turned the name/number equivalence into a game. Each letter of the alphabet is given a value from 1 to 26, A being 1 and Z 26. The names displayed in the game have a total value which consists of the sum of the values of their letters. Thus, *Abe* has a value of  $1+2+5=8$ . (Can you think of a name with a lower value than *Abe*?)

The game works in the following way: A name is displayed on the top of the screen and below it four other names are displayed. You have to figure out which one of the four has the same value as the first name displayed.

There is a strategy you might use to help you eliminate some names immediately. Look at the original name and see how many of its letters are skewed toward the front of the alphabet, how many are in the middle, and how many at the end. That should give you an idea of whether the value of the whole will be low, medium, or high. For example, it is clear that *Abe* will have a low value and *Val* a much higher one. With this first take, you can scan all the names and eliminate those whose values clearly don't match up.

```

100 REM *** KABALA NAME GAME ***
110 DIM A$(10),T$(10),K$(40),P(2),M(3)
:OPEN #1,4,0,"K:":GRAPHICS 0
200 POKE 752,1: ? CHR$(125):POSITION 10
,2: ? "THE KABALA NAME GAME":K$(1)=" ":
K$(40)=" ":K$(2)=K$:K=1:FOR T=0 TO 1
210 FOR J=0 TO 12:POSITION 4+T*6,6+J: ?
CHR$(64+K); "=";K:K=K+1:NEXT J:NEXT T
220 R=INT(RND(1)*20):R1=INT(RND(1)*4)
230 R2=INT(RND(1)*4):IF R2=R1 THEN 230
240 RESTORE 500+R:FOR T=0 TO R1:READ T
$:NEXT T:POSITION 25,6: ? T$
250 RESTORE 500+R:FOR T=0 TO R2:READ A
$:NEXT T:K$(31,40)=A$:FOR T=0 TO 2
260 P(T)=INT(RND(1)*20):FOR J=0 TO 2: I
F (P(T)=P(J) AND T<J) OR P(T)=R THEN
260
270 NEXT J:RESTORE 500+P(T):FOR J=0 TO
INT(RND(1)*4):READ T$:NEXT J:K$(T*10+
1,T*10+10)=T$:NEXT T:FOR T=0 TO 3
280 M(T)=4:NEXT T:FOR T=0 TO 3
290 M=INT(RND(1)*4):IF M(M)<4 THEN 290
300 M(M)=T:IF T=3 THEN P=M
310 NEXT T:FOR T=0 TO 3:POSITION 21,10
+T*2:PRINT T+1; ". ";K$(M(T)*10+1,M(T)*
10+10):NEXT T:POKE 764,255
320 GET #1,K:K=K-49:IF K<0 OR K>3 THEN
320
330 POSITION 19,10+K*2: ? "*":POSITION
24,10+P*2:FOR J=1 TO LEN(A$): ? CHR$(12
B+ASC(A$(J,J))):NEXT J
340 IF K<>P THEN FOR T=200 TO 250:SOUN
D 0,T,10,10:NEXT T:SOUND 0,0,0,0
350 IF K=P THEN FOR T=20 TO 0 STEP -1:
SOUND 0,T,10,10:SOUND 1,20-T,10,10:NEX
T T:SOUND 0,0,0,0:SOUND 1,0,0,0:X=X+1
360 Y=Y+1:POSITION 7,20: ? "YOU'VE GOTT
EN ";X;" OUT OF ";Y;"!":IF Y<10 THEN F
OR T=1 TO 1000:NEXT T:GOTO 200
370 POSITION 7,22: ? "PRESS START TO PL
AY AGAIN"

```

```

380 IF PEEK(53279)<>6 THEN 380
390 RUN
500 DATA EDDIE,ADELE,JAKE,DICK
501 DATA DIANA,TED,ANN,GAIL
502 DATA DEBRA,JANE,JOE,ALICE
503 DATA JIM,DAVE,NEAL,LEO
504 DATA DIANE,SAM,MABEL,FRED
505 DATA BILL,BETH,HEIDI,ERIC
506 DATA KATE,PAT,NICK,GREG
507 DATA DAVID,JOAN,NEIL,LINDA
508 DATA MARIA,JANICE,TIM,ALEX
509 DATA ANDY,RAY,HELEN,TINA
510 DATA DANIEL,SUE,PHIL,CLARK
511 DATA BECKY,PETE,BOBBY,KATIE
512 DATA RON,SARAH,JOHN,DOUG
513 DATA IRENE,PAULA,GARY,FREDDIE
514 DATA CATHY,GEORGE,MARY,NANCY
515 DATA ALBERT,DANNY,PEDRO,JESSE
516 DATA CHARLES,RUBY,MANUEL,MICKEY
517 DATA VICKY,HARRY,HENRY,SIMON
518 DATA TONY,SUSAN,PENNY,JOSHUA
519 DATA JEREMY,THOMAS,JERRY,HERBERT

```

Here are the data statements for the game. We've made lists of names of equivalent number value. You add or subtract names so long as all the names on a given line have the same value. You can also replace a line of your own with animal names or plant names and change the nature of the game that way. Just be sure to change the PRINT commands and check the numerical values of your words.

```

500 DATA EDDIE,ADELE,JAKE,DICK
501 DATA DIANA,TED,ANN,GAIL
502 DATA DEBRA,JANE,JOE,ALICE
503 DATA JIM,DAVE,NEAL,LEO
504 DATA DIANE,SAM,MABEL,FRED
505 DATA BILL,BETH,HEIDI,ERIC
506 DATA KATE,PAT,NICK,GREG
507 DATA DAVID,JOAN,NEIL,LINDA
508 DATA MARIA,JANICE,TIM,ALEX
509 DATA ANDY,RAY,HELEN,TINA
510 DATA DANIEL,SUE,PHIL,CLARK
511 DATA BECKY,PETE,BOBBY,KATIE
512 DATA RON,SARAH,JOHN,DOUG
513 DATA IRENE,PAULA,GARY,FREDDIE
514 DATA CATHY,GEORGE,MARY,NANCY
515 DATA ALBERT,DANNY,PEDRO,JESSE
516 DATA CHARLES,RUBY,MANUEL,MICKEY
517 DATA VICKY,HARRY,HENRY,SIMON
518 DATA TONY,SUSAN,PENNY,JOSHUA
519 DATA JEREMY,THOMAS,JERRY,HERBERT

```

# 10

## ***Pitch and Toss***

This game is thoroughly determined by chance.

Games of chance exist everywhere in the world. Often they are associated with predicting the future or making important decisions when there are no reasons to choose one way of acting over another. They also are good gambling games since every player is supposed to have an equal chance of winning. In our society, the most familiar chance games involve dice, playing cards, and spinners like the roulette wheel. In other parts of the world, different devices are used. For example, throughout the African continent cowrie shells, nuts, and elaborately carved wood tokens are commonly used in chance



games. The characteristic of these devices is that they have two sides like our coins. A number of them are cast and scores are made according to which faces show up.

For example, there is a game played by the Igbo of Nigeria which consists of casting four cowrie shells which can fall back up or open side up.



The best scores are four backs up or four backs down.

There are 16 possibilities when tossing the four cowries: (U = back up, D = back down)

UUUU	UUDD	DDUU
UUUD	UDUD	UDDD
UUDU	UDDU	DUDD
UDUU	DUUD	DDUD
DUUU	DUDU	DDDU
		DDDD

Since the cowries can be considered to look alike, these possibilities can be simplified in the following way:

4 up 0 down — 1 way  
 3 up 1 down — 4 ways  
 2 up 2 down — 6 ways  
 1 up 3 down — 4 ways  
 0 up 4 down — 1 way

Because of this, the following scoring system could be used for the game, giving fewer points to the arrangement most likely to happen, and more points to the rarer combinations.

4 up or 4 down	4 points
1 up 3 down	2 points

3 up 1 down

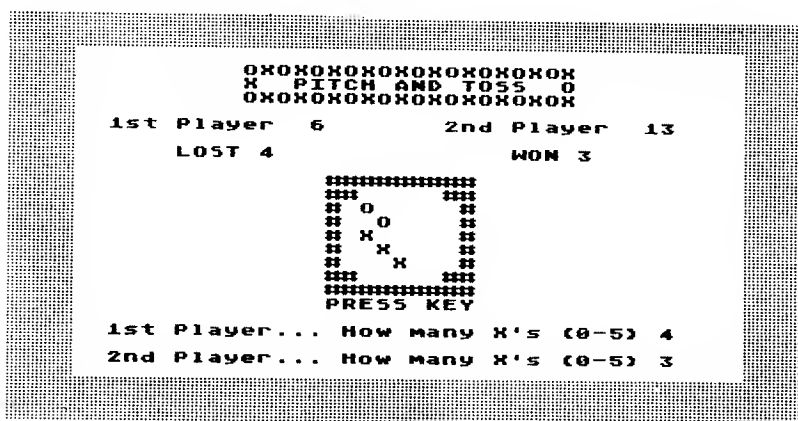
2 points

2 up 2 down

1 point

Try the game with cowries, walnuts, or pennies. It is as interesting as playing dice and provides more awareness of arrays of figures and scoring combinations.

This game is a computer version of the cowrie game. The computer will throw five cowries onto the playing board. One side of the cowrie will be represented by X and the other side as O. You have to guess how many X's will come up on your toss. The game is for two people and both players guess on each throw. The scoring is as follows: If you guess correctly you get points equal to the number of X's that you predicted would come up. If your partner guesses correctly, you lose points equal to the number of your guess. If neither of you guess correctly there is no score and you go on to the next round.



```
100 REM *** PITCH AND TOSS ***
110 GRAPHICS 0:POKE 752,1:POKE 82,10:P
RINT :PRINT "OXOXOXOXOXOXOXOXOX"
120 PRINT "X PITCH AND TOSS O":PRINT
"OXOXOXOXOXOXOXOXOX":OPEN #1,4,0,"K
:":S1=10:S2=10:GOSUB 600:GOSUB 700
200 POSITION 2,20:PRINT "1st Player...
How many X's (0-5) ";:GOSUB 400
210 PRINT K:G1=K:POSITION 2,22:PRINT "
2nd Player... How many X's (0-5) ";
220 GOSUB 400:IF K=G1 THEN 220
230 PRINT K:G2=K
```



```

240 C=0:FOR T=0 TO 4:X=INT(RND(1)*5):P
=INT(RND(1)*2):POSITION 17+X,11+T:PRIN
T CHR$(79+9*P):IF P THEN C=C+1
250 NEXT T:IF C=G1 THEN S1=S1+G1:S2=S2
-G2:POSITION 6,7:PRINT "WON ";G1:POSIT
ION 26,7:PRINT "LOST ";G2
260 IF C=G2 THEN S2=S2+G2:S1=S1-G1:POS
ITION 6,7:PRINT "LOST ";G1:POSITION 26
,7:PRINT "WON ";G2
270 GOSUB 600:IF S1<1 OR S2<1 THEN POS
ITION 15,18:PRINT "GAME OVER":PRINT CH
R$(253):GOSUB 810:RUN
280 GOSUB 800:FOR T=1 TO 3:POSITION 2,
20:PRINT CHR$(156):NEXT T:GOSUB 700:GO
TO 200
400 GET #1,K:K=K-48:IF K<0 OR K>5 THEN
400
410 RETURN
600 POSITION 2,5:PRINT "1st Player ";
S1;" ":POSITION 22,5:PRINT "2nd Player
";S2;" ":RETURN
700 POKE 82,15:POSITION 15,9:PRINT "##
#####":PRINT "##      ##":FOR T=1 TO
5:PRINT "*"      #":NEXT T
710 PRINT "##      ##":PRINT "#####
":RETURN
800 POSITION 15,18:PRINT "PRESS KEY"
810 POKE 764,255
820 IF PEEK(764)=255 THEN 820
830 POKE 764,255:POSITION 15,18:PRINT
"      ":POSITION 6,7:PRINT "
":POSITION 26,7:PRINT "      ":RETURN

```

# 11

## Date to Weekday

This is a conversion program. You can enter any date—past, present, or future—and this program will tell you which day of the week it happened to be. Leap years are also indicated. Using this program, you can find out the day you and your friends and family were born. You can find out the days Napoleon, Emma Goldman, and Julius Caesar were born. You can find out what day Christmas will fall on in 2100 or what day it fell on in 1166.

```

100 REM   *** DATE TO WEEKDAY ***
110 DIM D$(9),M$(9)
120 POKE 752,1:PRINT CHR$(125):POSITION
N 9,3:PRINT "DATE TO DAY CONVERSION":P
OSITION 2,6:PRINT "MONTH (1-12) ";
130 INPUT M:PRINT :PRINT "DAY (1-31)
";:INPUT D:PRINT
140 PRINT "YEAR (0- ) ";:INPUT Y:PR
INT :PRINT
200 YH=INT(Y/28):YL=Y-YH*28:RESTORE 30
0:FOR T=0 TO YL:READ YK:NEXT T
210 LP=0:IF INT(Y/4)=Y/4 THEN LP=1
220 N=0:K=0:RESTORE 400:FOR T=1 TO M:N
=N+K:READ K:NEXT T:N=N+D-1+YK:IF M>2 T
HEN N=N+LP
230 N=N-7*INT(N/7)
240 RESTORE 500:FOR T=0 TO N:READ D$:N
EXT T
250 RESTORE 600:FOR T=1 TO M:READ M$:N
EXT T:PRINT M$;" ";D$," ";Y;" WAS A ";
D$;"."
260 IF LP THEN PRINT :PRINT Y;" WAS A
LEAP YEAR."
270 POSITION 14,20:PRINT "PRESS START"
280 IF PEEK(53279)<>6 THEN 280
290 POKE 764,255:GOTO 120
300 DATA 4,6,0,1,2,4,5,6,0,2,3,4,5,0,1
,2,3,5,6,0,1,3,4,5,6,1,2,3
400 DATA 31,28,31,30,31,30,31,31,30,31
,30,31
500 DATA MONDAY,TUESDAY,WEDNESDAY,THUR
SDAY,FRIDAY,SATURDAY,SUNDAY
600 DATA JANUARY,FEBRUARY,MARCH,APRIL,
MAY,JUNE,JULY,AUGUST,SEPTEMBER,OCTOBER
,NOVEMBER,DECEMBER

```

# 12

## President Precedence

For multiple choice fans, here's a challenge to test your knowledge of the Presidential terms of office. A date appears at the top of the screen. The names of three Presidents appear by the numbers one, two, and three. Press a number on the keyboard to match the correct President with his term.

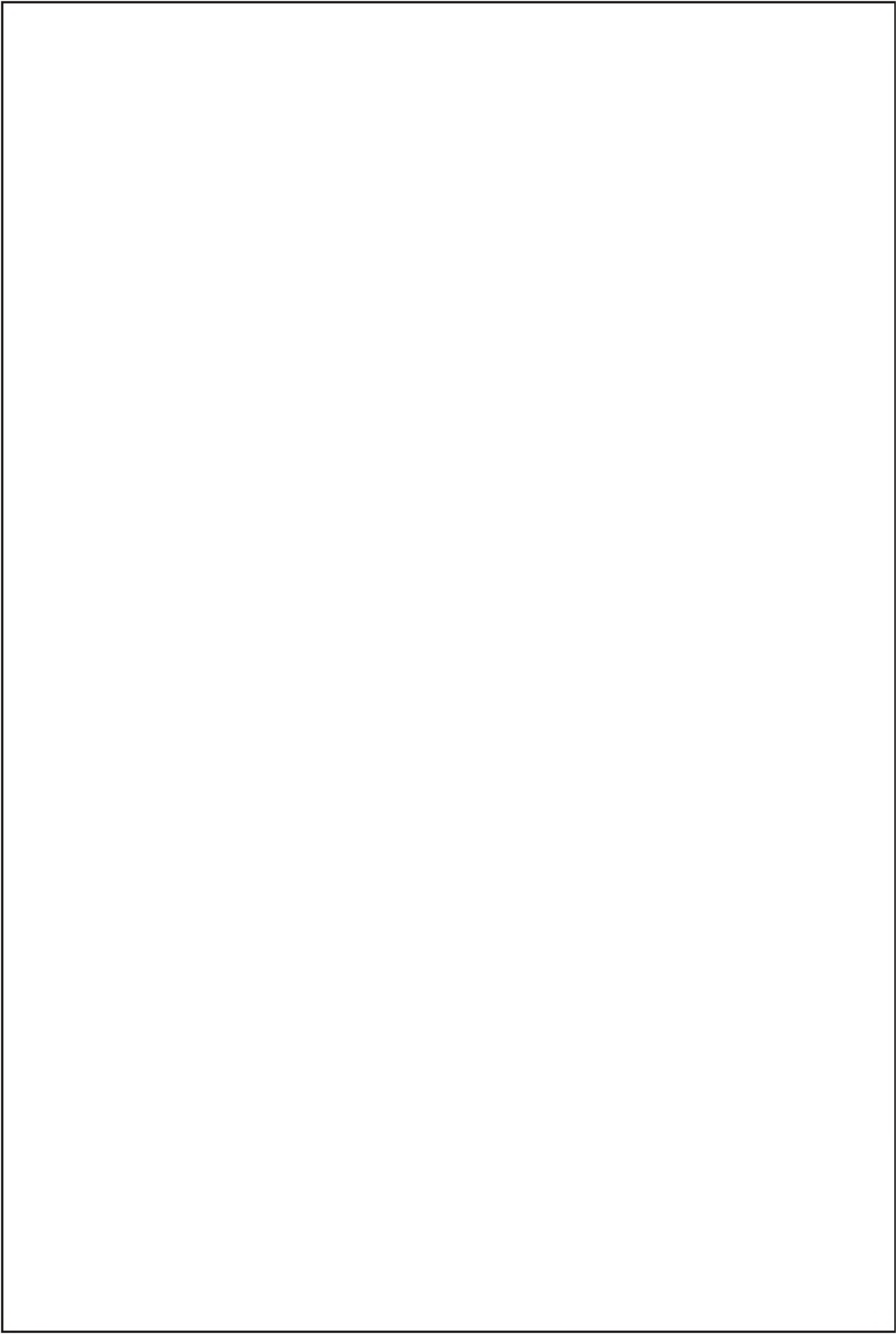
```

100 REM ** PRESIDENT PRECEDENCE **
110 DIM T$(30),N$(1200),Y$(360),P(2),M
(2):OPEN #1,4,0,"K:":N$(1)=" ":N$(1200)
)=" ":N$(2)=N$:Y$=N$:FOR T=0 TO 39
120 READ T$:N$(1+T*30,30+T*30)=T$:READ
T$:Y$(1+T*9,9+T*9)=T$:NEXT T
130 P(0)=INT(RND(1)*40):P(1)=INT(RND(1)
)*40:P(2)=INT(RND(1)*40):IF P(1)=P(0)
OR P(2)=P(0) OR P(2)=P(1) THEN 130
140 POKE 752,1:PRINT CHR$(125):POSITIO
N 10,3:? "PRESIDENT PRECEDENCE":POSITIO
N 15,6:? Y$(1+P(0)*9,9+P(0)*9)
150 FOR T=0 TO 2:M(T)=3:NEXT T:POSITIO
N 2,10:FOR T=0 TO 2:R=INT(RND(1)*3):IF
M(R)<3 THEN T=T-1:NEXT T
160 M(R)=T:NEXT T:FOR T=0 TO 2:K=P(M(T)
):PRINT T+1;". ";N$(1+K*30,30+K*30):N
EXT T
170 GET #1,K:IF K<49 OR K>51 THEN 170
180 IF M(K-49)<>0 THEN 170
190 POSITION 16,18:PRINT "CORRECT":FOR
T=1 TO 500:NEXT T:GOTO 130
500 DATA George Washington,1789-1797,J
ohn Adams,1797-1801,Thomas Jefferson,1
801-1809,James Madison,1809-1817
510 DATA James Monroe,1817-1825,John Q
uincy Adams,1825-1829,Andrew Jackson,1
829-1837,Martin Van Buren,1837-1841
520 DATA William Henry Harrison,1841-1
841,John Tyler,1841-1845,James K. Polk
,1845-1849,Zachary Taylor,1849-1850
530 DATA Millard Fillmore,1850-1853,Fr
anklin Pierce,1853-1857,James Buchanan
,1857-1861,Abraham Lincoln,1861-1865
540 DATA Andrew Johnson,1865-1869,Ulys
ses S. Grant,1869-1877,Rutherford B. H
ayes,1877-1881,James A. Garfield
550 DATA 1881-1881,Chester A. Arthur,1
881-1885,Grover Cleveland (1),1885-188
9,Benjamin Harrison,1889-1893
560 DATA Grover Cleveland (2),1893-189
7,William McKinley,1897-1901,Theodore
Roosevelt,1901-1909,William H. Taft
570 DATA 1909-1913,Woodrow Wilson,1913
-1921,Warren G. Harding,1921-1923,Calv
in Coolidge,1923-1929
580 DATA Herbert C. Hoover,1929-1933,F
ranklin D. Roosevelt,1933-1945,Harry S
. Truman,1945-1953
590 DATA Dwight D. Eisenhower,1953-196
1,John F. Kennedy,1961-1963,Lyndon B.
Johnson,1963-1969,Richard M. Nixon

```

600 DATA 1969-1974,Gerald R. Ford,1974  
-1977,Jimmy Carter,1977-1981,Ronald Re  
agan,1981-????

This program can be modified to match events to dates. For example, in the data lines from 500 to 600 you can put the name of a battle with its dates or the name of a great invention with the date it was invented. As long as you do not change the structure of the data and keep to associating a name with a number you will be able to modify this program and make up your own multiple choice quizzes. There is one thing to be careful about, however. On line 120, the program is set up to deal with 40 data inputs (FOR T=0 to 39) and on line 130 to deal with the length of your DATA statement. If you keep to the same number of items and same length of items in the program you'll have no trouble. If not, you should change the parameters on lines 120 and 130.



# PART III

---

## IQ Building



**I**ntelligence Quotient (IQ) implies that there is some objective measure of how smart people are. An IQ of 100 is supposed to be normal, 140 brilliant, and 75 slow. However, no one knows how accurate even the most sophisticated tests are. Your IQ has to do with how you perform on the tests you take and not with how intelligently you function with others and at your work.

Since the measure of IQ is so test-bound, your score can be changed if you practice the skills that the test designers have chosen as epitomizing the essence of "intelligence." Usually, these skills have to do with memorizing lists, figuring out sequences, estimating answers quickly, and knowing how to manipulate strings of symbols. This Part provides a number of games and exercises that hone these skills. Of course, there is no guarantee that your IQ will be raised by playing them but there is a chance that, if you master these modes of thinking, you will find it easier to deal with the challenges of IQ tests.

# 13

## Gestalt

The word *Gestalt* refers to a perceptual whole. When we look at a chair or table we do not see its individual parts and then conclude that they go together to make a chair or table. We see the chair or table as an integrated whole or Gestalt. A Gestalt problem that English philosophers debated about 30 years ago led to this computer program. The problem was usually phrased this way:

You are looking at a field of sheep. Exactly how many sheep do you see? If a sheep were added or subtracted would you see any more or fewer sheep?

If you do not stop to count the sheep, it would be accurate to say that you saw a whole bunch of sheep, not a particular number of sheep. You could, however, estimate the number of sheep and, if you had a great deal of experience with sheep, could get more and more accurate. Estimation even within the context of an initially disordered Gestalt can be refined and improved. This program gives you an opportunity to guess at the number of dots on your computer screen without counting them. The computer will tell you how close your guess is and then give you another series of dots to guess about. This kind of estimation is related to IQ tests in that the ability to estimate the probability of an answer makes it possible to eliminate the least likely choices of a multiple choice question and, therefore, gives you more time to work on problems.

The program also helps you concentrate on responding to your perceptions and refining them. It is practice in looking at and interpreting what you see.

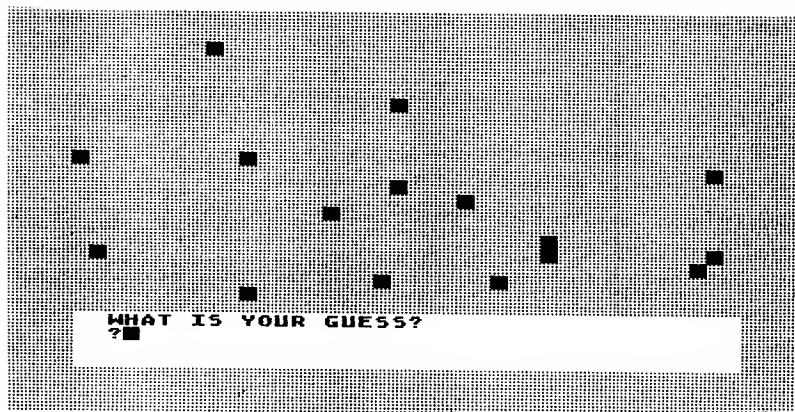
```
100 REM      *** GESTALT ***
110 GRAPHICS 0:POKE 752,1:PRINT :PRINT
  "THIS IS AN ESTIMATION GAME.":PRINT
120 PRINT "YOU WILL SEE DOTS FLASH ON
  THE SCREEN":PRINT
140 PRINT :PRINT "DO NOT TRY TO COUNT
  THE DOTS!!":PRINT :PRINT
```



```

150 PRINT "THE COMPUTER WILL TELL YOU
HOW CLOSE":PRINT
160 PRINT "YOUR GUESS IS. AFTER A WHI
LE YOU":PRINT
170 PRINT "SHOULD GET BETTER AT ESTIMA
TING THEIR":PRINT :PRINT "NUMBER."
180 PRINT :PRINT "PRESS SPACEBAR TO BE
GIN"
190 IF PEEK(764)<>33 THEN 190
200 POKE 764,255:GRAPHICS 3:COLOR 1
210 LET Q=INT(RND(1)*25)
220 FOR Z=1 TO Q
230 LET X=INT(RND(1)*40)
240 LET Y=INT(RND(1)*20)
250 PLOT X,Y
260 NEXT Z
270 PRINT "WHAT IS YOUR GUESS?"
280 INPUT S
290 IF S=Q THEN PRINT "YOU GOT IT.":FO
R T=1 TO 1500:NEXT T:GOTO 110
300 IF S<Q THEN PRINT "YOU ARE ";Q-S;"
TOO LOW":FOR T=1 TO 1500:NEXT T:GOTO
110
310 IF Q<S THEN PRINT "YOU ARE ";S-Q;"
TOO HIGH":FOR T=1 TO 1500:NEXT T:GOTO
110

```



This program may put too many dots on the screen at the same time or run too quickly for you. If you want fewer dots on the screen to begin with, you can change the value of Q on line 210:

```
210 LET Q=INT(RND(1)*25)
```

For example, if you change the 25 at the end of the line to 5, you'll get a maximum of 5 dots and if you change it to 10 you'll get a maximum of 10.

If you want to slow the program down, you can put a pause loop such as

```
FOR X=1 TO 500:NEXT X
```

into the program after you plot the points (line 250). This line could be numbered 255 so it would come after the plot command and yet remain within the loop.

# 14

---

## **Flash Line**

Flash Line is a length estimating program. It flashes two lines on the screen and asks you which one is longer. If you get the correct answer, it gives you another set of lines. If you get it wrong, it gives you another chance. Most of the challenges of this program are easy to answer once you get the rhythm of looking and inputting your answer. However, the game can be speeded up by changing the pause loops at lines 170 and 210. You can also change the length of the lines by reducing 160 in lines 110 and 120. When you reduce the size of the line the game gets more complex because the sizes of the lines don't vary as much.

Notice that line 130 eliminates the possibility of the computer displaying two equal lines. The routine of which this is a part can be very useful when you want to develop games that generate unequal quantities. Here is that routine:

```
110 LET X=INT(RND(1)*160)
120 LET Y=INT(RND(1)*160)
130 IF X=Y THEN GOTO 110
```

In effect, line 130 returns the program to line 110 if X and Y are the same; then new random numbers are chosen.

```
100 REM   *** FLASH LINE ***
110 LET X=INT(RND(1)*160)
120 LET Y=INT(RND(1)*160)
130 IF X=Y THEN GOTO 110
140 GRAPHICS 7:COLOR 1
150 GOSUB 600
160 PLOT 0,30:DRAWTO X,30
170 FOR A=1 TO 300:NEXT A
180 GRAPHICS 7
190 GOSUB 600
200 PLOT 0,60:DRAWTO Y,60
210 FOR B=1 TO 300:NEXT B
220 GRAPHICS 0
230 GOSUB 600
240 PRINT "TYPE 1 OR 2 "
250 INPUT Z
260 IF Z=1 THEN GOTO 400
270 IF Z=2 THEN GOTO 500
400 IF X>Y THEN PRINT "RIGHT. HERE'S
ANOTHER ONE.":FOR C=1 TO 500:NEXT C:GO
TO 110
410 IF Y>X THEN PRINT "TAKE ANOTHER LO
OK.":FOR C=1 TO 500:NEXT C:GOTO 140
500 IF Y>X THEN PRINT "RIGHT. HERE'S
ANOTHER ONE.":FOR C=1 TO 500:NEXT C:GO
TO 110
510 IF X>Y THEN PRINT "TAKE ANOTHER LO
OK.":FOR C=1 TO 500:NEXT C:GOTO 140
600 PRINT "WHICH LINE IS LONGER,THE 1s
t OR 2nd?":RETURN
```

# 15

---

## Did You See It?

Here is a third estimating game. This uses the Atari Graphics 2 mode to print big stars. The program moves the stars around the screen in a random way using the POSITION statement. An interesting percep-

tual experiment would be to see whether substituting other symbols for \* changes the difficulty of the challenge. For example at line 180:

```
180 POSITION X,Y:PRINT #6;"*"
```

try to change the "\*" to "?" or "m" or "\$" or any other symbol and see whether the nature of the challenge changes.

```
100 REM *** DID YOU SEE IT? ***
110 GRAPHICS 2:COLOR 1
120 POKE 752,1:PRINT
130 S=INT(RND(1)*10)+1
140 PRINT CHR$(125):PRINT "HOW MANY ST
ARS DO YOU SEE?"
150 FOR M=1 TO 500:NEXT M
160 FOR W=1 TO S
170 X=INT(RND(1)*20):Y=INT(RND(1)*10)
180 POSITION X,Y:PRINT #6;"*"
190 FOR R=1 TO 75:NEXT R
200 POSITION X,Y:PRINT #6;" "
210 NEXT W
220 INPUT G
230 IF G=S THEN PRINT "YOU GOT IT!":FO
R M=1 TO 500:NEXT M:GOTO 130
240 IF G<>S THEN PRINT "TRY AGAIN.":FO
R M=1 TO 500:NEXT M:GOTO 140
```

# 16

---

## Memory Muscle Builder

This challenge begins by showing you two digits to memorize. If you can type them in correctly you will then be shown three digits and so on until your mind is overcome with numbers. When you finally miss you will be assigned one of the 15 ranks of memory power we have built into the program. Repeating number sequences occur on a number of individually administered IQ tests and, with some training, you can become a memory master. The memorization of sequences is also a good exercise in the development of concentration.

```

100 REM * MEMORY MUSCLE BUILDER *
110 DIM A(15),T$(30):OPEN #1,4,0,"K:":
N=1
120 GRAPHICS 0:POKE 752,1:POSITION 7,2
:PRINT "* MEMORY MUSCLE BUILDER *"
130 POSITION 2,6:PRINT "WATCH CAREFULL
Y...":FOR T=1 TO 150:NEXT T
140 POSITION 2,8:FOR T=0 TO N:A(T)=INT
(RND(1)*10):PRINT A(T);" ";:NEXT T:FOR
T=1 TO 100+N*300:NEXT T
150 POSITION 2,8:FOR T=0 TO N:PRINT "?
";:NEXT T
160 POSITION 2,10:PRINT "NOW, ENTER TH
E NUMBERS YOU SAW":POSITION 2,12
170 FOR T=0 TO N
180 GET #1,K:K=K-48
190 IF K<0 OR K>9 THEN 180
200 PRINT K;" ";:IF K<>A(T) THEN 300
210 NEXT T:POSITION 2,14:PRINT "WOW, W
HAT A MEMORY!":N=N+1:IF N=16 THEN 400
220 FOR T=1 TO 750:NEXT T:GOTO 120
300 POSITION 2,14:PRINT "OOPS! THE NU
MBERS WERE..."
310 POSITION 2,16:FOR T=0 TO N:PRINT A
(T);" ";:NEXT T
320 RESTORE 500+N-1:READ T$:POSITION 2
,18:PRINT "YOUR RANK: ";T$:FOR T=1 TO
2000:NEXT T:RUN
400 POSITION 2,16:PRINT "INCREDIBLE!
YOU MUST BE A COMPUTER!":PRINT :END
500 DATA NOVICE CLASS 2
501 DATA NOVICE CLASS 1
502 DATA INITIATE CLASS 2
503 DATA INITIATE CLASS 1
504 DATA AVERAGE CLASS 2
505 DATA AVERAGE CLASS 1
506 DATA MEMORY WHIZ CLASS 2
507 DATA MEMORY WHIZ CLASS 1
508 DATA MASTER CLASS 2
509 DATA MASTER CLASS 1
510 DATA MEMORY MASTER CLASS 2
511 DATA MEMORY MASTER CLASS 1
512 DATA GRANDMASTER
513 DATA PRODIGY
514 DATA GENIUS

```

If you look carefully at this program you will see that the variable N indicates how many numbers you guessed correctly and at line 320, is used to assign you a rank. The assignment of ranks depends

upon a READ...DATA... command in which each rank becomes a single data statement as listed below:

```
500 DATA NOVICE CLASS 2
501 DATA NOVICE CLASS 1
502 DATA INITIATE CLASS 2
503 DATA INITIATE CLASS 1
504 DATA AVERAGE CLASS 2
505 DATA AVERAGE CLASS 1
506 DATA MEMORY WHIZ CLASS 2
507 DATA MEMORY WHIZ CLASS 1
508 DATA MASTER CLASS 2
509 DATA MASTER CLASS 1
510 DATA MEMORY MASTER CLASS 2
511 DATA MEMORY MASTER CLASS 1
512 DATA GRANDMASTER
513 DATA PRODIGY
514 DATA GENIUS
```

If you want to create your own ranks, all you have to do is list the program and change the data lines to any titles you choose. If you want to create a subroutine in another program that uses a ranking system, all you have to do is adapt lines 320 to 514 for your own purposes.

# 17

---

## Mug Shot

Mug Shot is a different kind of memory game. Mug shots are face photos that victims of crime are given in order to identify criminals. In this program, the computer creates a face for you to look at for a while and then clears the screen and challenges you to recreate the face from a choice of eyes, noses, and mouths. After you think you have successfully reconstructed the face, the computer shows you the original face next to your reconstruction. This program provides a deceptively complex challenge and gives a sense of how hard it is to reconstruct a face from memory. You can study this program and add



a number of other features to the face. For example, you can put on eyebrows or hair, or even make a whole series of different shaped faces (oval, round, squarish, etc.).

This program uses CONTROL Graphics in Graphics 0 to create the faces. It, therefore, doesn't use DRAWTO or PLOT statements. Instead of printing out each CONTROL character, it uses the CHR\$(X) function which uses the ASCII numbers for each symbol on your Atari keyboard. Consult Appendix C of your *ATARI BASIC Reference Manual* for a chart of ASCII numbers.

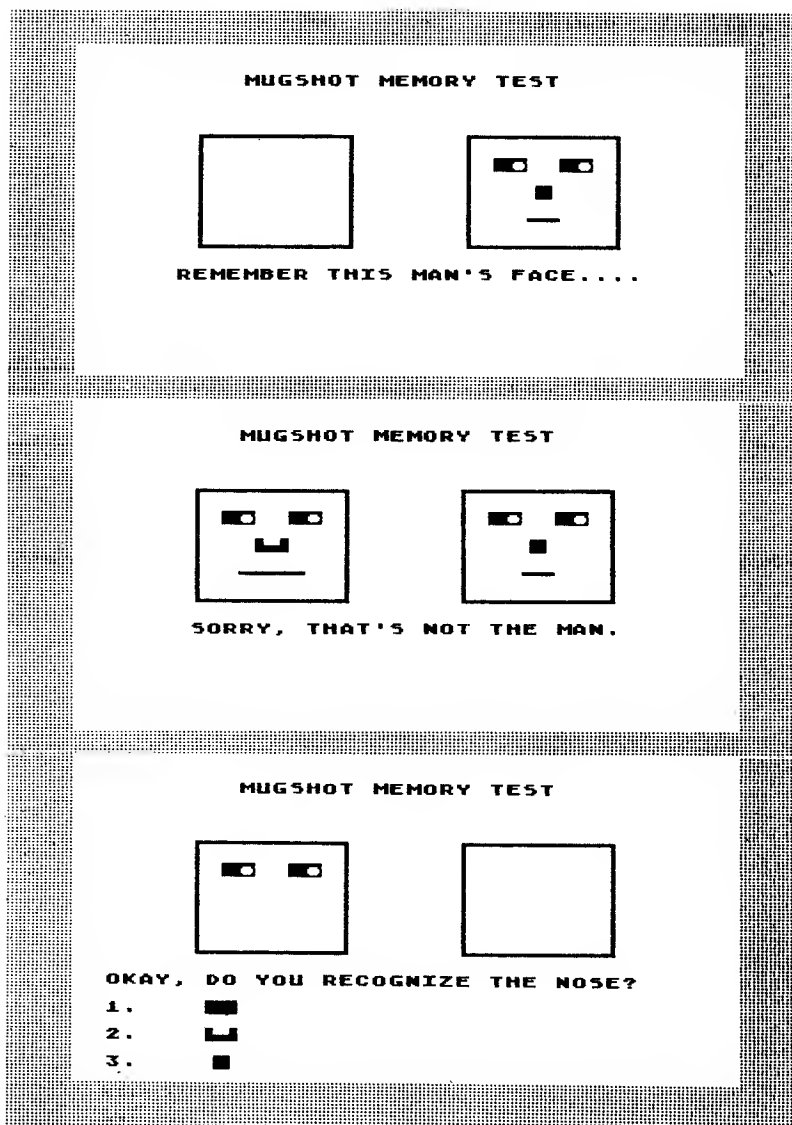
Here is a simple way to understand how CHR\$(X) and the ASCII codes for graphics characters work. You'll notice on the chart that the triangle that is printed on the screen by typing in [CTRL][H] has the ASCII number 136. You can have your computer print this character in two ways:

```
PRINT [CTRL][H]
```

or

```
PRINT CHR$(136)
```

If you look at the Mug Shot program you will find many uses for CHR\$ that you can understand by using the ASCII chart. Also, as an exercise, try to draw one of the faces in the program using CHR\$ statements.





```

100 REM      *** MUG SHOTS ***
110 GRAPHICS 0:POKE 752,1:PRINT
120 DIM G$(54),K$(5),A(2),B(2),M(2),M$
(3):FOR T=1 TO 54:READ K:G$(T,T)=CHR$(
K):NEXT T:OPEN #1,4,0,"K:"
200 PRINT CHR$(125):POSITION 10,2:PRIN
T "MUGSHOT MEMORY TEST"
210 GOSUB 600:GOSUB 610:FOR T=0 TO 2:A
(T)=INT(RND(1)*3):NEXT T:GOSUB 500
220 POSITION 6,16:? "REMEMBER THIS MAN
'S FACE....":FOR T=1 TO 750:NEXT T:GOS
UB 610:GOSUB 700:FOR T=0 TO 2:Z=M(T)
230 POSITION 2,16:? "OKAY, DO YOU RECO
GNIZE THE ";:RESTORE 890+Z:READ K$:? K
$;"? ":FOR J=0 TO 2:POKE 82,2
240 POSITION 2,18+J*2:PRINT J+1;". ";
G$(Z*18+J*6+1,Z*18+J*6+6):NEXT J
250 GET #1,K:IF K<49 OR K>51 THEN 250
260 B(Z)=K-49:POSITION 9,8+2*Z:PRINT G
$(Z*18+B(Z)*6+1,Z*18+B(Z)*6+6):NEXT T
270 FOR T=1 TO 7:POSITION 2,16:? CHR$(
156):NEXT T:GOSUB 500:FOR T=0 TO 2:IF
A(T)=B(T) THEN NEXT T:GOTO 300
280 POSITION 7,16:? "SORRY, THAT'S NOT
THE MAN.":FOR T=200 TO 250 STEP 2:SOU
ND 0,T,10,10:NEXT T:GOTO 400
300 POSITION 13,16:? "THAT'S THE MAN!"
:FOR T=1 TO 5:FOR J=20 TO 0 STEP -1:SO
UND 0,J,10,10:NEXT J:NEXT T
400 SOUND 0,0,0,0:POKE 764,255
410 IF PEEK(764)=255 THEN 410
420 POKE 764,255:GOTO 200
500 FOR T=0 TO 2:POSITION 25,8+2*T:PRI
NT G$(T*18+A(T)*6+1,T*18+A(T)*6+6):NEX
T T:RETURN
600 POKE 82,7:POSITION 7,6:GOSUB 900:R
ETURN
610 POKE 82,23:POSITION 23,6:GOSUB 900
:RETURN
700 M$="      ":FOR T=0 TO 2
710 R=INT(RND(1)*3):S=R+1:IF M$(S,S)="
*" THEN 710
720 M$(S,S)="*":M(T)=R:NEXT T:RETURN
800 DATA 8,10,32,32,8,10,32,20,32,32,2
0,32,160,148,32,32,160,148
810 DATA 32,32,160,160,32,32,32,32,139
,140,32,32,32,32,153,25,32,32
820 DATA 32,18,18,18,18,32,32,26,18,18
,3,32,32,32,18,18,32,32
890 DATA EYES
891 DATA NOSE
892 DATA MOUTH

```

```

900 ? CHR$(17);:FOR T=1 TO 8:? CHR$(18
):NEXT T:? CHR$(5):FOR T=1 TO 7:? CHR
$(124);:FOR J=1 TO 8:? " ";:NEXT J
910 ? CHR$(124):NEXT T:? CHR$(26);:FOR
T=1 TO 8:? CHR$(18);:NEXT T:? CHR$(3)
:RETURN

```

# 18

## Numbers Runner

In this game you are presented with a grid consisting of 20 stars as well as 20 blanks at the bottom of the screen. The numbers from 1 to 20 have been hidden under the stars randomly. The game is played with a joystick in port 1. Using the joystick, you can move around the grid. When you press the button on the joystick a number will appear under the star you are currently on. Your goal is to place the numbers on the blank spaces at the bottom of the screen in order from 1 to 20. When you find 1, press the button and a 1 will appear in the first blank on the upper left. When you find 2 press the button and a 2 will appear in the blank to the right of 1. You have to remember what is under the stars that you have tested out in order to get all 20 numbers before your time runs out. There is a timer built into this program as well as a score keeper. The timer sets the length of a turn to one minute. The scorer will record your highest score.

This is a memory development game—one that involves the kinds of skills called upon when you are asked about details of stories you have read or pictures you have seen. It is possible, after a while, to register all of the numbers that you land on and solve the puzzle in half the time you are allowed.

The timer in the program is controlled by two lines, 160 and 290. There are three memory locations in your Atari 800 that control what is called a *jiffy clock* that can be used for many different programs. They are locations 18, 19, and 20. A *jiffy* is 1/60th of a second. Location 20 counts up to 256 jiffies or about 4 seconds. Each 256 jiffies (or four seconds) is recorded in memory location 19 which can count 256\*4 seconds. Memory location 18 records cycles of memory

location 19 and, therefore, units of  $4 \times 256 \times 256$  seconds. You can think of location 20 as counting seconds, 19 as counting hours, and 18 as counting days. The reason the number 256 is involved is that the Atari 800 is an 8-bit byte machine; that is, each memory location is one unit (called a *byte*) of eight on/off switches. Using base 2 these eight switches can store up to 256 different numbers.

Now take a look at line 290. It reads:

```
290 IF PEEK(19) > 15 THEN GOTO 500
```

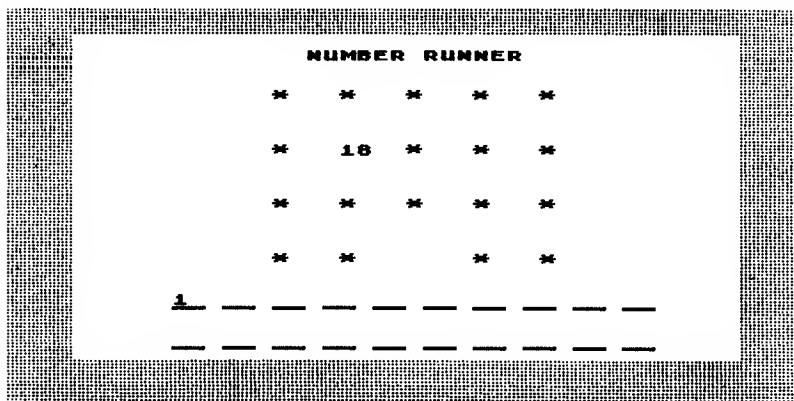
PEEK(19) tells you how many cycles of 4 seconds' length each location 19 has recorded. Fifteen cycles of approximately 4 seconds duration give you just about  $15 \times 4$  or 60 seconds—the one minute the timer allows. When location 19 goes beyond the recorded minute, the program jumps to line 500 which stops the game and prints out your score.

If you want to change the timer to 2 minutes, all you have to do is rewrite line 290 as follows:

```
290 IF PEEK(19) > 30 THEN GOTO 500
```

Try it and also try to speed up the game by setting location 19 to 30 seconds.

The other line involved with the keeping is line 160. It resets the clock to 0 before you begin a new game. These two lines can be used in any programming context where you want to control the time.



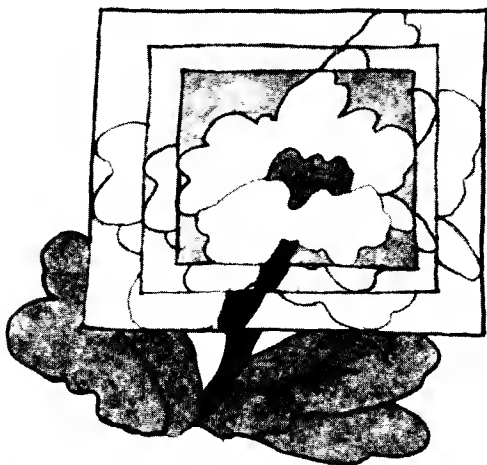
```

100 REM      *** NUMBER RUNNER ***
110 DIM M$(20),PX(4,3),PY(4,3),N(4,3):
GRAPHICS 0:POKE 752,1:SETCOLOR 2,12,4:
SETCOLOR 4,12,4:SETCOLOR 1,0,12
120 PRINT CHR$(125):POSITION 14,1:PRIN
T "NUMBER RUNNER":M$="0000000000000000
0000":FOR X=0 TO 4:FOR Y=0 TO 3
130 N=INT(RND(1)*20)+1:IF M$(N,N)="1"
THEN 130
140 M$(N,N)="1":N(X,Y)=N:PX(X,Y)=X*4+1
2:PY(X,Y)=Y*4+4:POSITION PX(X,Y),PY(X,
Y):PRINT "*":NEXT Y:NEXT X
150 FOR Y=20 TO 23 STEP 3:FOR X=0 TO 2
7 STEP 3:POSITION 6+X,Y:PRINT CHR$(13)
;CHR$(13);" ":NEXT X:NEXT Y
160 POKE 19,0:POKE 20,0:X=2:Y=1:RX=3:R
Y=19:R=1
180 IF STRIG(0)=0 AND N(X,Y)>0 THEN 40
0
190 T=STICK(0):IF T=15 THEN 250
200 IF T>8 AND T<12 THEN X=X-1:IF X<0
THEN X=4
210 IF T>4 AND T<8 THEN X=X+1:IF X>4 T
HEN X=0
220 IF T=10 OR T=14 OR T=6 THEN Y=Y-1:
IF Y<0 THEN Y=3
230 IF T=9 OR T=13 OR T=5 THEN Y=Y+1:I
F Y>3 THEN Y=0
240 FOR T=36 TO 0 STEP -4:SOUND 0,T,10
,12:NEXT T
250 POSITION PX(X,Y),PY(X,Y):IF N(X,Y)
>0 THEN PRINT " ":GOTO 270
260 PRINT "0"
270 FOR T=1 TO 50:NEXT T:IF N(X,Y)>0 T
HEN POSITION PX(X,Y),PY(X,Y):PRINT "*"
:GOTO 290
280 POSITION PX(X,Y),PY(X,Y):PRINT " "
290 IF PEEK(19)>15 THEN 500
300 GOTO 180
400 POSITION PX(X,Y),PY(X,Y):PRINT N(X
,Y):GOSUB 600
410 IF STRIG(0)=1 THEN 410
420 IF N(X,Y)<>R THEN POSITION PX(X,Y)
,PY(X,Y):PRINT "* ":GOSUB 600:GOTO 180
430 RX=RX+3:IF RX>34 THEN RX=6:RY=RY+3
440 POSITION RX,RY:PRINT R:R=R+1:POSIT
ION PX(X,Y),PY(X,Y):PRINT " ":FOR T=2
55 TO 0 STEP -5:SOUND 0,T,10,12
450 NEXT T:SOUND 0,0,0,0:N(X,Y)=0:IF R
<21 THEN GOSUB 600:GOTO 180
500 POSITION 14,17:PRINT "YOUR SCORE:
";R-1:IF R-1>HS THEN HS=R-1

```

```
510 POSITION 14,18:PRINT "HIGH SCORE:"  
";HS:GOSUB 600:FOR T=1 TO 200:NEXT T  
520 IF STRIG(0)=0 THEN 120  
530 GOTO 520  
600 IF STRIG(0)=0 THEN 600  
610 RETURN
```

## A Pause for Meditation



# 19

---

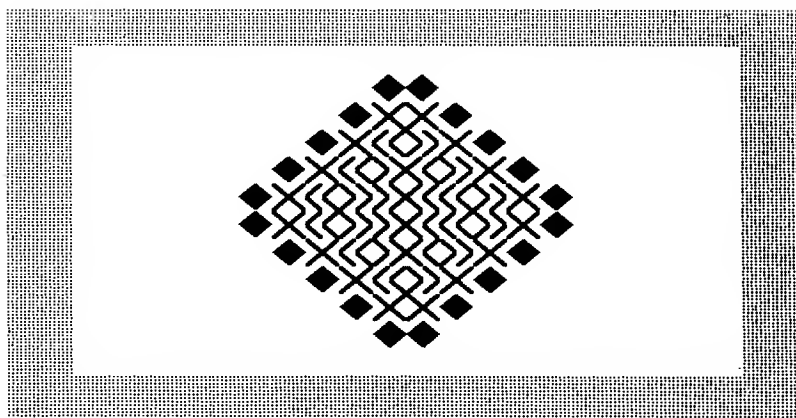
### *Meditative Images*

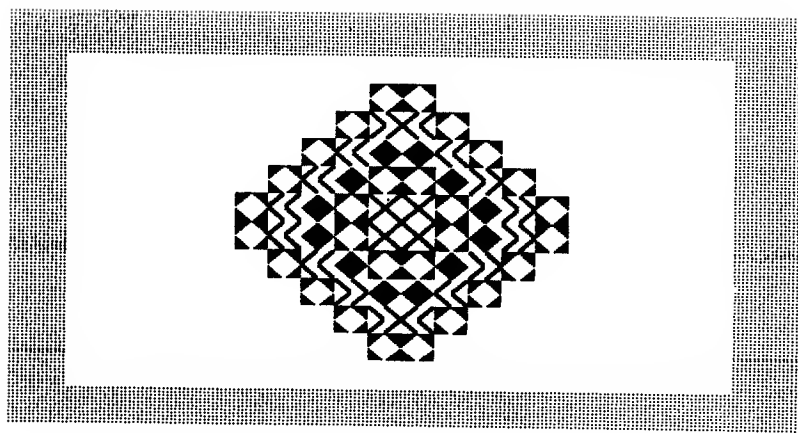
Here is a program that you can relax with. It generates different colors and images that show off the use of [CRTL] graphics and the SET-COLOR function of the Atari and provide a relaxing, slowly changing visual experience to savor.

```

100 REM *** MEDITATIVE IMAGES ***
110 GRAPHICS 0:POKE 752,1:PRINT :SETCO
LOR 2,0,0:POKE 82,0:DIM I(8)
200 FOR Y=0 TO 8 STEP 2
210 LI=INT(RND(1)*5)
220 IF LI=I(Y) THEN 210
230 I(Y)=LI:RI=LI:IF LI>2 THEN RI=7-LI
240 FOR X=0 TO Y STEP 2
250 POSITION 18-X,10-Y+X:GOSUB 500+LI
260 POSITION 18-X,12+Y-X:GOSUB 500+LI
270 POSITION 20+X,10-Y+X:GOSUB 500+RI
280 POSITION 20+X,12+Y-X:GOSUB 500+RI
290 NEXT X:FOR T=1 TO 350:NEXT T:NEXT
Y
300 SETCOLOR 2,INT(RND(1)*16),2:SETCOL
OR 1,0,INT(RND(1)*9)+6:GOTO 200
500 PRINT CHR$(8);CHR$(10);CHR$(30);CH
R$(30);CHR$(29);CHR$(138);CHR$(136):RE
TURN
501 PRINT CHR$(136);CHR$(138);CHR$(30)
;CHR$(30);CHR$(29);CHR$(10);CHR$(8):RE
TURN
502 PRINT CHR$(7);CHR$(6);CHR$(30);CHR
$(30);CHR$(29);CHR$(6);CHR$(7):RETURN
503 PRINT CHR$(7);CHR$(7);CHR$(30);CHR
$(30);CHR$(29);CHR$(6);CHR$(6):RETURN
504 PRINT CHR$(6);CHR$(6);CHR$(30);CHR
$(30);CHR$(29);CHR$(7);CHR$(7):RETURN

```





The SETCOLOR function built into Atari BASIC gives you considerable control over the 128 colors that the Atari can generate. It has three variables:

### SETCOLOR A,B,C

The A variable refers to the color register. In the case of this program, which is in Graphics 0, if A=2 the background color can be changed and if A=1 the foreground color (letters, numbers, and graphics characters) can be changed. For more details on how to use the color registers in all of the Graphics modes, consult your *ATARI BASIC Reference Manual*.

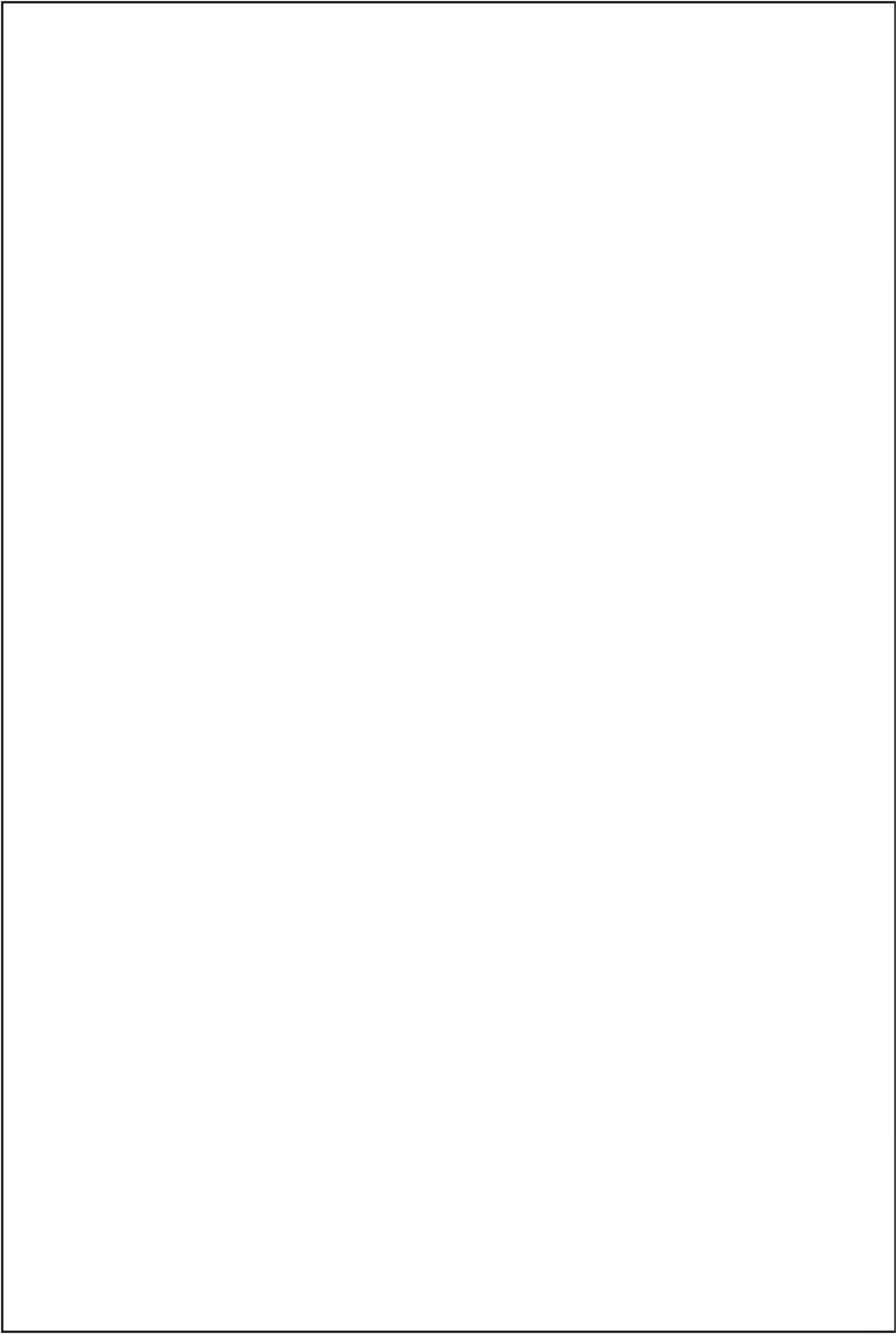
The B variable controls the color and ranges from 0 to 15 while the C variable controls the luminance of the color and ranges from 0 to 14 (the higher the brighter). It is the combination of different colors and luminances that generates the pallet of the Atari 800.

Here is a short program that shows you how the color changes work in the Meditative Imagery program. Notice at line 40 that first the background color is changed by using SETCOLOR 2,C,D and then the foreground is changed by using SETCOLOR 1,B,C.

```
10 GRAPHICS 0:SETCOLOR 2,0,0
20 PRINT "HHHHHHHHH"
30 FOR X=1 TO 500:NEXT X
```

```
40 SETCOLOR 2, INT(RND(1)*16),2:SETCOLOR  
1,0,INT(RND(1)*9)+6  
50 PRINT "YYYYYYYYY"  
60 FOR X=1 TO 500:NEXT X  
70 GOTO 10
```





# PART IV

---

## Strategy Puzzles



**T**he games and puzzles in this chapter require some thought. Their outcome is not decided by a spinner, a roll of dice, or any other chance device. If you work at them you should be able to figure out a way to solve their challenges (or, in one case, prove that a solution is impossible) and even beat the computer which has been programmed to play against you. If you read through the commentary on the games you'll also find out how to add a timer to your own games as well as some other information about game challenges and programming strategies.

# 20

## The Pattern Game

This game requires cooperation in an indirect way. You are given a 4 x 4 grid and asked to make a pattern on it. Using a joystick, you can create a star pattern on it. Using a joystick, you can create a star pattern by moving the cursor to any spot on the grid where you would like to place a star. If you press the button, a star will appear on your grid. Press SELECT when you are finished and your pattern will be covered over. Have a friend try to guess your pattern by moving the joystick and pressing the button to place a star where he or she thinks it might be. Here's where the cooperation comes in. If you don't create a regular pattern on the grid, your friend will certainly have a hard time reconstructing your grid. In fact, if you simply place stars at random on the grid, the game will be no fun at all. You have to create a pattern that is challenging and decipherable. By the way, if your friend is getting frustrated he or she can press the OPTION key to get a clue. One of the stars will be revealed.

Here are a few challenging patterns:

```
* * * *
* " " *
* " " *
* * * *
```

```
* * " "
" " * *
* * " "
" " * *
```

```
" * " *
* " * "
" * " *
* " * "
```

**THE PATTERN GAME**  
**MAKE YOUR HIDDEN PATTERN...**

```

* * * *
* * * *
* * * *
* * * *

```

Press SELECT when you've finished.

```

100 REM *** THE PATTERN GAME ***
110 DIM C(3),P1(3,3),P2(3,3):C(1)=96:C
(2)=42:C(3)=88:GRAPHICS 0:POKE 752,1
120 ? CHR$(125):POSITION 12,3:? "THE P
ATTEN GAME":POSITION 7,5
130 ? "MAKE YOUR HIDDEN PATTERN...":PO
KE 82,9:POSITION 9,8:K=1:GOSUB 700
140 POSITION 3,19:? "Press SELECT when
you've finished.":X=0:Y=0:CL=0
200 P=STRIG(0):IF P=0 THEN P1(X,Y)=P1(
X,Y)+1:IF P1(X,Y)>3 THEN P1(X,Y)=1
210 IF P=0 THEN POSITION 9+X*2,8+Y*2:?
CHR$(C(P1(X,Y))):GOSUB 550
220 GOSUB 600:POSITION 9+X*2,8+Y*2:? "
":IF X<>ZX OR Y<>ZY THEN FOR T=10 TO
0 STEP -1:SOUND 0,T,10,8:NEXT T
230 GOSUB 500:POSITION 9+X*2,8+Y*2:? C
HR$(C(P1(X,Y))):GOSUB 500
240 IF PEEK(53279)<>5 THEN 200
300 POSITION 4,5:? "NOW GUESS THE HIDD
EN PATTERN...":POKE 82,9:POSITION 9,8
310 K=3:GOSUB 700:POKE 82,24:POSITION
24,8:K=2:GOSUB 700:POSITION 3,21:? "Pr
ess OPTION to get a clue.":X=0:Y=0
400 P=STRIG(0):IF P=0 THEN P2(X,Y)=P2(
X,Y)+1:IF P2(X,Y)>3 THEN P2(X,Y)=1
410 IF P=0 THEN POSITION 24+X*2,8+Y*2:
? CHR$(C(P2(X,Y))):GOSUB 550
420 GOSUB 600:POSITION 24+X*2,8+Y*2:?
" ":IF X<>ZX OR Y<>ZY THEN FOR T=10 TO
0 STEP -1:SOUND 0,T,10,8:NEXT T
430 GOSUB 500:POSITION 24+X*2,8+Y*2:?
CHR$(C(P2(X,Y))):GOSUB 500
440 IF PEEK(53279)=3 THEN 900
450 IF PEEK(53279)<>5 THEN 400

```

```

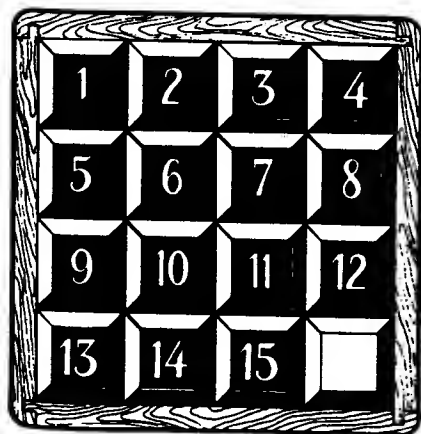
460 FOR J=0 TO 3:FOR T=0 TO 3:IF ABS(P
1(T,J))<>P2(T,J) THEN ? CHR$(253):GOTO
400
470 NEXT T:NEXT J:POSITION 1,5:? "YOU
DID IT WITH ";CL;" CLUES! PRESS START
":FOR J=1 TO 3:FOR T=20 TO 0 STEP -1
480 SOUND O,T,10,10:NEXT T:NEXT J
490 IF PEEK(53279)<>6 THEN 490
495 GOTO 120
500 FOR T=1 TO 10:IF STICK(O)<>15 OR S
TRIG(O)=0 THEN T=10
510 NEXT T:RETURN
550 IF STRIG(O)=0 THEN 550
560 RETURN
600 Z=STICK(O):ZX=X:ZY=Y:IF Z>8 AND Z<
12 THEN X=X-1:IF X<0 THEN X=0
610 IF Z>4 AND Z<8 THEN X=X+1:IF X>3 T
HEN X=3
620 IF Z=10 OR Z=14 OR Z=6 THEN Y=Y-1:
IF Y<0 THEN Y=0
630 IF Z=9 OR Z=13 OR Z=5 THEN Y=Y+1:I
F Y>3 THEN Y=3
640 RETURN
700 FOR J=0 TO 3:FOR T=0 TO 3:IF K=2 T
HEN ? CHR$(C(P2(T,J)))"; "
710 IF K=1 THEN P1(T,J)=1:P2(T,J)=1:?
CHR$(96);" "
720 IF K=3 THEN ? ". ";
730 NEXT T:? :? :NEXT J:RETURN
900 IF P1(X,Y)<0 THEN 400
910 POSITION 9+X*2,8+Y*2:? CHR$(C(P1(X
,Y))):POSITION 24+X*2,8+Y*2:? CHR$(C(P
1(X,Y))):CL=CL+1:P2(X,Y)=P1(X,Y)
920 P1(X,Y)=-P1(X,Y):GOTO 400

```

## 21, 22

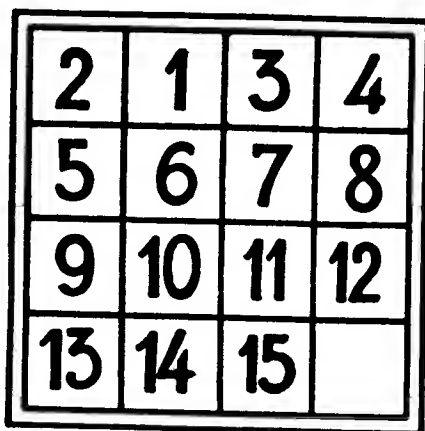
### 3 x 3 and 4 x 4 Number Puzzlers

The 3x3 and 4x4 Number Puzzlers are computer versions of puzzles that are over 100 years old. The earliest versions of the puzzles consisted of wooden trays with numbered wood blocks laid into them as in this old drawing:

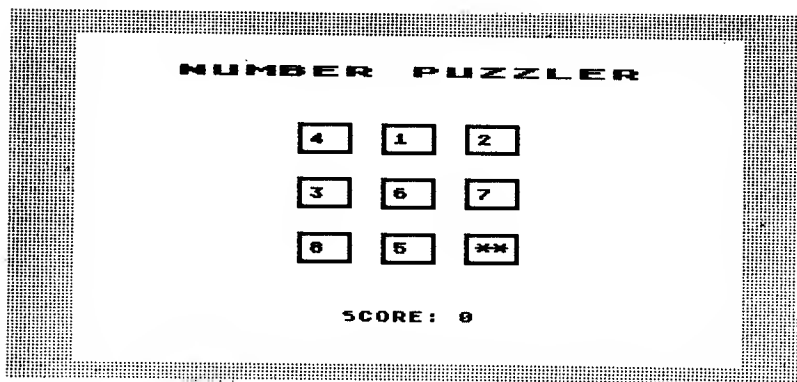
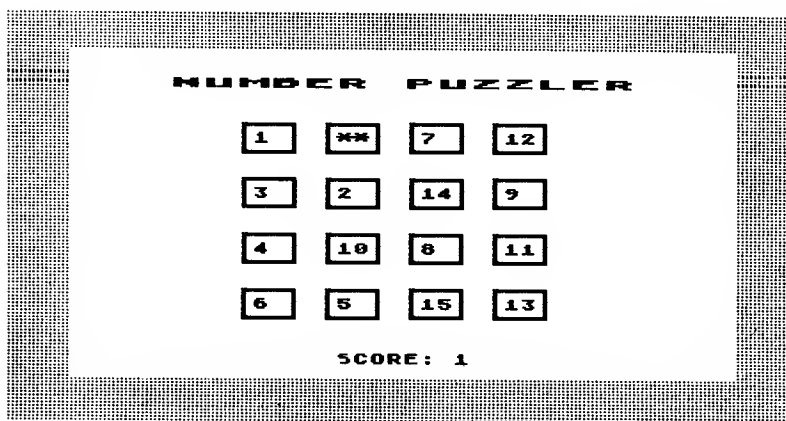


The 4x4 puzzle consists of a 4x4 square grid with the numbers 1 through 15 printed on the blocks. There is one blank square on the grid. The blocks can be moved onto the empty square and, little by little, shuffled around the entire grid. The puzzle consists of shifting the numbers around the grid and then reassembling them in consecutive order. Here is one possible challenge on the 4x4 board:

Put this square in order so that 1 is at the top left and all the other numbers proceed in order from left to right and then top to bottom as in the first drawing:



The computer versions of the program first use a 3x3 board and the numbers 1 to 8, and then a 4x4 board and numbers 1 to 15. The computer shuffles the numbers for you and gives you a board to solve. The empty square is marked by double stars (\*\*). Using a joystick, you can move the empty square vertically or horizontally onto any adjoining square. The number in that square will then move to the previously empty square. Thus, in the computer version you are moving the empty square and the numbers are switched for you. The net result, however, is identical to moving the blocks into the empty square. There is a counter built into the program that will let you know how many numbers you have in the right place.





```

100 REM      *** 3 BY 3 PUZZLER ***
110 DIM P(2,2),A$(9),B$(26):FOR T=1 TO
  26:READ K:B$(T,T)=CHR$(K):NEXT T
120 DATA 17,18,18,5,29,30,30,30,30,124
  ,32,32,124,29,30,30,30,30,26,18,18,3,2
  8,30,30,30
130 GRAPHICS 0:SETCOLOR 2,0,0:POKE 752
  ,1:PRINT CHR$(125)
140 T=PEEK(560)+256*PEEK(561):POKE T+7
  ,6:POKE T+8,6:POKE T+29,65:POKE T+30,P
  EEK(560):POKE T+31,PEEK(561)
150 POSITION 3,2:PRINT "number puzzler
  "

160 A$="0000000000"
170 FOR Y=0 TO 2:FOR X=0 TO 2
180 T=INT(RND(1)*9):IF A$(T+1,T+1)="1"
  THEN 180
190 POSITION 13+X*5,5+Y*4:PRINT B$;:IF
  T<>0 THEN PRINT T:GOTO 210
200 PRINT "":PX=X:PY=Y
210 A$(T+1,T+1)="1":P(X,Y)=T:NEXT X:NE
  XT Y:GOTO 310
220 T=STICK(0):IF T=15 OR T=5 OR T=6 O
  R T=9 OR T=10 THEN 220
230 OX=PX:OY=PY
240 IF T=14 THEN PY=PY-1:IF PY<0 THEN
  PY=0:GOTO 220
250 IF T=13 THEN PY=PY+1:IF PY>2 THEN
  PY=2:GOTO 220
260 IF T=11 THEN PX=PX-1:IF PX<0 THEN
  PX=0:GOTO 220
270 IF T=7 THEN PX=PX+1:IF PX>2 THEN P
  X=2:GOTO 220
280 POSITION 13+PX*5,5+PY*4:PRINT B$;"
  *"
290 POSITION 13+OX*5,5+OY*4:PRINT B$;P
  (PX,PY)
300 P(OX,OY)=P(PX,PY):P(PX,PY)=0
310 FOR K=0 TO 1:FOR T=10 TO 0 STEP -2
  :SOUND 0,T,10,10:NEXT T:NEXT K
320 C=1:FOR Y=0 TO 2:FOR X=0 TO 2
330 IF P(X,Y)=0 THEN 360
340 IF P(X,Y)<>C THEN 370
350 C=C+1
360 NEXT X:NEXT Y
370 POSITION 16,19:PRINT "SCORE: ";C-1
  :GOTO 220

```

```

100 REM      *** 4 BY 4 PUZZLER ***
110 DIM P(3,3),A$(16),B$(26):FOR T=1 T
O 26:READ K:B$(T,T)=CHR$(K):NEXT T
120 DATA 17,18,18,5,29,30,30,30,30,124
,32,32,124,29,30,30,30,30,26,18,18,3,2
B,30,30,30
130 GRAPHICS 0:SETCOLOR 2,0,0:POKE 752
,1:PRINT CHR$(125)
140 T=PEEK(560)+256*PEEK(561):POKE T+7
,6:POKE T+8,6:POKE T+29,65:POKE T+30,P
EEK(560):POKE T+31,PEEK(561)
150 POSITION 3,2:PRINT "number puzzler
"

160 A$="000000000000000000"
170 FOR Y=0 TO 3:FOR X=0 TO 3
180 T=INT(RND(1)*16):IF A$(T+1,T+1)="1
" THEN 180
190 POSITION 10+X*5,4+Y*4:PRINT B$;IF
T<>0 THEN PRINT T:GOTO 210
200 PRINT "":PX=X:PY=Y
210 A$(T+1,T+1)="1":P(X,Y)=T:NEXT X:NE
XT Y:GOTO 310
220 T=STICK(0):IF T=15 OR T=5 OR T=6 O
R T=9 OR T=10 THEN 220
230 OX=PX:OY=PY
240 IF T=14 THEN PY=PY-1:IF PY<0 THEN
PY=0:GOTO 220
250 IF T=13 THEN PY=PY+1:IF PY>3 THEN
PY=3:GOTO 220
260 IF T=11 THEN PX=PX-1:IF PX<0 THEN
PX=0:GOTO 220
270 IF T=7 THEN PX=PX+1:IF PX>3 THEN P
X=3:GOTO 220
280 POSITION 10+PX*5,4+PY*4:PRINT B$;"
*"
290 POSITION 10+OX*5,4+OY*4:PRINT B$;P
(PX,PY)
300 P(OX,OY)=P(PX,PY):P(PX,PY)=O
310 FOR K=0 TO 1:FOR T=10 TO 0 STEP -2
:SOUND O,T,10,10:NEXT T:NEXT K
320 C=1:FOR Y=0 TO 3:FOR X=0 TO 3
330 IF P(X,Y)=0 THEN 360
340 IF P(X,Y)<>C THEN 370
350 C=C+1
360 NEXT X:NEXT Y
370 POSITION 16,21:PRINT "SCORE: ";C-1
;" ":GOTO 220

```



There is a slight problem with this puzzle. Not all mixes of numbers can be unscrambled. There are some positions that are impossible to move into consecutive order. Sam Lloyd, one of the great puzzle inventors of all time, discovered several such positions in the 1870's and in one of his books describes how he turned this impossibility into a new challenge:

Older inhabitants of Puzzleland will remember how in the seventies I drove the entire world crazy over a little box of movable blocks which became known as the "14-15 Puzzle." The fifteen blocks were arranged in the square box in regular order, but with the 14 and 15 reversed as shown in the above illustration. The puzzle consisted of moving the blocks about, one at a time, to bring them back to the present position in every respect except that the error in the 14 and 15 was corrected.

A prize of \$1,000, offered for the first correct solution to the problem, has never been claimed, although there are thousands of persons who say they performed the required feat.

People became infatuated with the puzzle and ludicrous tales are told of shopkeepers who neglected to open their stores; of a distinguished clergyman who stood under a street lamp all through a wintry night trying to recall the way he had performed the feat. The mysterious feature of the puzzle is that none seem to be able to remember the sequence of moves whereby they feel sure they succeeded in solving the puzzle. Pilots are said to have wrecked their ships, and engineers rush their trains past stations. A famous Baltimore editor tells how he went for his noon lunch and was discovered by his frantic staff long past midnight pushing little pieces of pie around on a plate! Farmers are known to have deserted their plows, and I have taken one such instance as an illustration for the sketch.

Several new problems which developed from the original puzzle are worth giving:

Second Problem—Start again with the blocks as shown in the large illustration and move them so as to get the numbers in regular order, but with the vacant square at upper left-hand corner instead of lower right-hand corner (see Fig. 1).

Third Problem—Start with the blocks as before, turn the box a quarter-way round and move the blocks until they rest as in Fig. 2.

Fourth Problem—Start as before, then shift the pieces until they form a "magic square," the numbers adding to thirty along all vertical and horizontal rows, and the two diagonals.

	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Fig. 1

4	8	12	
3	7	11	15
2	6	10	14
1	5	9	13

Fig. 2

Following are the answers to Lloyd's other challenges, taken from Dover Books' *Mathematical Puzzles of Sam Lloyd*, edited by Martin Gardner whose commentary provides the answers to Lloyd's challenges.

[The original puzzle is impossible to solve except by such skulduggery as turning the 6 and 9 blocks upside down. One of the

puzzle's peculiarities is that any such interchange, involving two blocks, immediately converts the puzzle to a solvable one. In fact, any odd number of interchanges has the same effect, whereas an even number leaves the puzzle unsolvable as before. Readers interested in learning something about the interesting mathematical structure underlying this puzzle are referred to the classic analysis by W. W. Johnson and W. E. Story in their article, "Notes on the 15-Puzzle," *American Journal of Mathematics*, Vol. 2, 1879, p. 397f, and to briefer discussions of the puzzle in standard references on recreational mathematics.—M.G.]

The other three problems are solved as follows:

Fig. 1 can be reached in 44 moves: 14, 11, 12, 8, 7, 6, 10, 12, 8, 7, 4, 3, 6, 4, 7, 14, 11, 15, 13, 9, 12, 8, 4, 10, 8, 4, 14, 11, 15, 13, 9, 12, 4, 8, 5, 4, 8, 9, 13, 14, 10, 6, 2, 1.

Fig. 2 can be reached in 39 moves: 14, 15, 10, 6, 7, 11, 15, 10, 13, 9, 5, 1, 2, 3, 4, 8, 12, 15, 10, 13, 9, 5, 1, 2, 3, 4, 8, 12, 15, 14, 13, 9, 5, 1, 2, 3, 4, 8, 12.

The magic square can be produced in 50 moves: 12, 8, 4, 3, 2, 6, 10, 9, 13, 15, 14, 12, 8, 4, 7, 10, 9, 14, 12, 8, 4, 7, 10, 9, 6, 2, 3, 10, 9, 6, 5, 1, 2, 3, 6, 5, 3, 2, 1, 13, 14, 3, 2, 1, 13, 14, 3, 12, 15, 3.

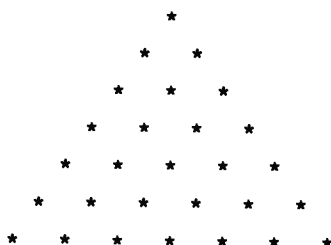
Try to figure out what characterizes an impossible challenge and certainly don't quit on the puzzles too soon. Something that may initially look impossible may merely be difficult.

## 23

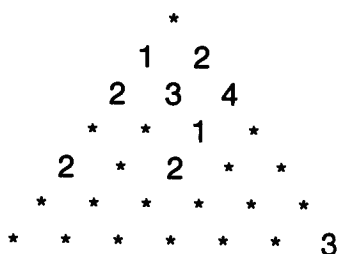
### Dozo

Dozo is a game that Japanese GO masters use to warm up. It is deceptively simple and involves a different sort of strategy than usually adopted in the more aggressive games of chess and checkers. Dozo requires that you play to not lose rather than play to win. This will become clear from a description of the board, pieces, and goals of the game.

Dozo is played on a board that is an equilateral triangle. In our version of the game, each side is seven units long. Here is a picture of the board:

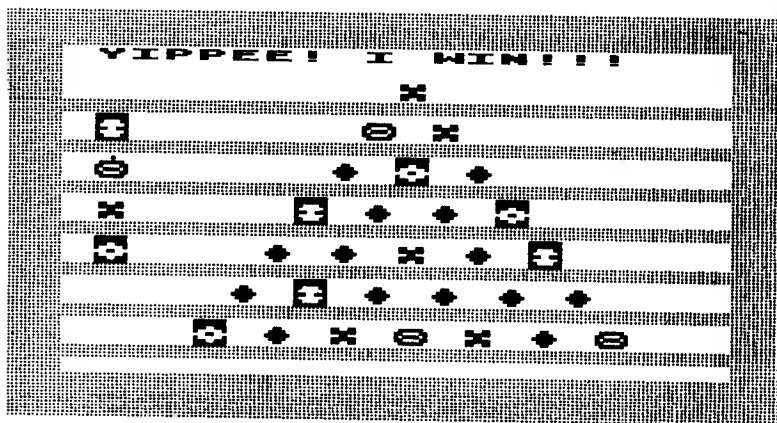
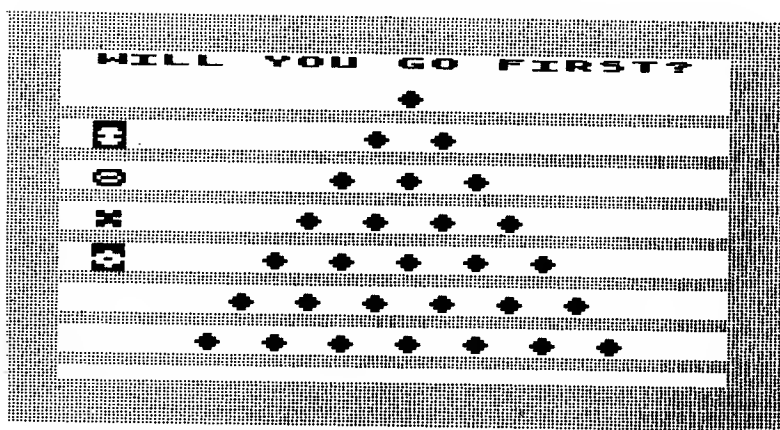


There are four different pieces that you can use in the game. As opposed to most Western games, the pieces do not belong to any one of the players. On each move of the game, you can choose any one of the pieces and place it on any empty point on the board. The goal of the game is to be the first player to make an equilateral triangle of any size with the three vertices of the triangle having the same piece. Suppose the pieces were called 1, 2, 3, and 4. Here is a sample game winning position:



Notice that the 2's in lines 3 and 5 make up an equilateral triangle. The player who placed the last 2 on the board is the winner. The player who placed the second 2 on the board trapped him- or herself into a losing position. The complexity of this game emerges from the fact that you always have to think about not making a move that will give your opponent an equilateral triangle. This involves analyzing positions that won't lead to a loss as well as looking out for winning positions. Simple and direct aggression will almost always lead to a loss in this game.

The Dozo program is the longest in this book and the most tedious to copy because it turns the computer into your opponent. Strategy for playing a good game of Dozo is built into the program and the computer is pretty hard to beat. However, the computer is not guaranteed a win and no one has yet analyzed Dozo fully enough to know what the best possible move is in any situation. Try your hand at crossing up the computer. It will take time and concentration. You will need a joystick to play this game.



```

100 REM          *** DOZO ***
1000 GOSUB 8000:IF C<>17076 THEN 9500
2000 GOSUB 6000:GOSUB 6200
2100 GOSUB 6300:IF FP THEN 3000
2110 POSITION 1,0:? #6;"my turn"
2200 IF CC(P)<=1 OR CC(P)=7 THEN 2400
2210 FOR T=0 TO CC-1:M1=M(P,CC):M2=M(P
,T):F1=M1:F2=M2:IF M1>M2 THEN F1=M2:F2
=M1
2220 B=B(F1)+(F2-(F1+1))*2:P1=ASC(A$(B
,B)):P2=ASC(A$(B+1,B+1))
2230 IF P1<=27 THEN IF S(P1)=ASC("/")
THEN W1=P1:GOTO 2300
2240 IF P2<=27 THEN IF S(P2)=ASC("/")
THEN W1=P2:GOTO 2300
2250 NEXT T:GOTO 2400
2300 GOSUB 6300:POSITION 1,0:? #6;"YIP
PEE! I WIN!!":X=X(SX(W1),SY(W1)):Y=Y(
SX(W1),SY(W1))
2310 FOR T=1 TO 2:POSITION X,Y:? #6;T$
:FOR J=0 TO 120:NEXT J:POSITION X,Y:?
#6;J$:FOR J=0 TO 80:NEXT J:NEXT T
2320 POSITION X,Y:? #6;T$:X1=X(SX(F1),
SY(F1)):Y1=Y(SX(F1),SY(F1)):X2=X(SX(F2
),SY(F2)):Y2=Y(SX(F2),SY(F2))
2330 FOR T=1 TO 5:POSITION X,Y:? #6;J$
:POSITION X1,Y1:? #6;J$:POSITION X2,Y2
:? #6;J$:FOR J=0 TO 80:NEXT J
2340 POSITION X,Y:? #6;T$:POSITION X1,
Y1:? #6;T$:POSITION X2,Y2:? #6;T$:FOR
J=0 TO 120:NEXT J:NEXT T
2350 FOR T=1 TO 4000:NEXT T:GOTO 2000
2400 P=0:IF CC(1)<CC(P) THEN P=1
2410 IF CC(2)<CC(P) THEN P=2
2420 IF CC(3)<CC(P) THEN P=3
2430 EP=P:PC=-1
2440 TT=0:TC=-1
2500 POP :IF TT<10 THEN 2520
2505 TC=TC+1:T=TC:IF T>27 THEN 6400
2510 IF S(T)<>ASC("/") THEN 2505
2515 GOTO 2535
2520 T=INT(RND(1)*28):IF S(T)<>ASC("/")
) THEN 2520
2525 IF CC(P)=0 THEN 2600
2530 TT=TT+1
2535 FOR J=0 TO CC(P)-1:M1=M(P,J):M2=T
:F1=M1:F2=M2:IF M1>M2 THEN F1=M2:F2=M1
2540 B=B(F1)+(F2-(F1+1))*2:P1=ASC(A$(B
,B)):P2=ASC(A$(B+1,B+1))
2545 IF P1<=27 THEN IF S(P1)=ASC("/")
THEN 2500
2550 IF P2<=27 THEN IF S(P2)=ASC("/")
THEN 2500

```



```

2555 NEXT J
2600 T$=P$(P+1,P+1)
2610 X=X(SX(T),SY(T)):Y=Y(SX(T),SY(T))
2620 FOR K=1 TO 5:POSITION X,Y:? #6;T$
:FOR J=0 TO 50:NEXT J:POSITION X,Y:? #
6;J$:FOR J=0 TO 50:NEXT J
2630 NEXT K:POSITION X,Y:? #6;T$
2640 S(T)=ASC(T$):PX(T)=X(SX(T),SY(T))
:PY(T)=Y(SX(T),SY(T)):M(P,CC(P))=T:CC(
P)=CC(P)+1
2700 FP=1:GOTO 2100
3000 POSITION 1,0:? #6;"YOUR TURN"
3100 IF STRIG(0)=0 AND CC(P)<7 THEN X=
0:Y=0:GOSUB 6100:GOTO 3500
3110 Z=STICK(0):IF Z=14 THEN P=P-1:IF
P<0 THEN P=3
3120 IF Z=13 THEN P=P+1:IF P>3 THEN P=
0
3130 POSITION 1,P+3:? #6;" "
3140 T$=P$(P+1,P+1)
3150 FOR T=0 TO 27:IF STRIG(0)=0 THEN
POSITION 1,P+3:? #6;T$:GOTO 3100
3160 NEXT T:POSITION 1,P+3:? #6;T$:FOR
T=0 TO 27:IF STRIG(0)=0 THEN 3100
3170 NEXT T:GOTO 3100
3500 IF STRIG(0)=0 THEN GOSUB 6100:GOT
O 3800
3510 Z=STICK(0):OLDX=X:OLDY=Y
3520 IF Z=14 THEN X=X-1:Y=Y-2
3530 IF Z=13 THEN X=X+1:Y=Y+2
3540 IF Z=7 THEN X=X+1
3550 IF Z=11 THEN X=X-1
3560 IF Z=6 THEN Y=Y-1
3570 IF Z=9 THEN Y=Y+1
3580 IF Z=5 THEN X=X+1:Y=Y+1
3590 IF Z=10 THEN X=X-1:Y=Y-1
3600 IF X<0 OR Y>6 OR X>Y THEN X=OLDX:
Y=OLDY
3610 SS=P(X,Y):J$=CHR$(S(SS)):XS=X(X,Y
):YS=Y(X,Y)
3620 POSITION XS,YS:? #6;" ":FOR T=0 T
O 10:NEXT T
3630 POSITION XS,YS:? #6;T$
3640 FOR T=0 TO 27:IF STRIG(0)=0 THEN
POSITION XS,YS:? #6;J$:GOTO 3500
3650 NEXT T:POSITION XS,YS:? #6;J$:FOR
T=0 TO 27:IF STRIG(0)=0 THEN 3500
3660 NEXT T:GOTO 3500
3800 IF S(SS)<>ASC("/") THEN 3510
3810 S(SS)=ASC(T$):POSITION XS,YS:? #6
;T$:PX(SS)=XS:PY(SS)=YS

```

```

3820 M(P,CC(P))=SS:CC=CC(P):CC(P)=CC(P)
+1:IF CC(P)<3 THEN FP=0:GOTO 2100
3830 FOR T=0 TO CC-2:M1=M(P,CC):M2=M(P
,T):F1=M1:F2=M2:IF M1>M2 THEN F1=M2:F2
=M1
3840 B=B(F1)+(F2-(F1+1))*2:P1=ASC(A$(B
,B)):P2=ASC(A$(B+1,B+1))
3850 FOR J=T+1 TO CC-1:IF P1=M(P,J) OR
P2=M(P,J) THEN 4000
3860 NEXT J:NEXT T:FP=0:GOTO 2100
4000 GOSUB 6300
4010 POSITION 1,0:? #6;"you won the ga
me!"
4020 FOR T=0 TO 4000:NEXT T:GOTO 2000
6000 ? #6;CHR$(125):FOR X=0 TO 6:FOR Y
=0 TO 6:IF Y<X THEN NEXT Y
6010 POSITION X(X,Y),Y(X,Y):? #6;"/":N
EXT Y:NEXT X:FOR T=3 TO 6:POSITION 1,T
:? #6;P$(T-2,T-2):NEXT T
6020 FOR T=0 TO 27:S(T)=ASC("/"):NEXT
T:FOR T=0 TO 3:CC(T)=0:NEXT T:RETURN
6100 IF STRIG(0)=0 THEN 6100
6110 RETURN
6200 POSITION 1,0:? #6;"WILL YOU GO FI
RST?"
6210 GET #1,K:FP=V0:IF K=89 THEN FP=1:
RETURN
6220 IF K=78 THEN RETURN
6230 GOTO 6210
6300 POSITION 0,0:? #6;"
":RETURN
6400 PC=PC+1:P=PC:IF P<4 THEN 2440
6410 T=INT(RND(1)*28):IF S(T)<>ASC("/"
) THEN 6410
6420 P=EP:GOTO 2600
8000 POKE 106,120:GRAPHICS 17:FOR T=0
TO 31:READ K:POKE 1536+T,K:NEXT T:T=US
R(1536,57344,31744):POKE 756,124
8010 DATA 104,104,133,204,104,133,203,
104,133,206,104,133,205,162,4,160,0,17
7,203,145,205,200,208,249,230,204
8020 DATA 230,206,202,208,240,96
8030 DIM A$(756),T$(1),J$(1),P$(4),S(2
7),PX(27),PY(27),B(26),CC(3),P(6,6),X(
6,6),Y(6,6),M(3,6),SX(27),SY(27)
8040 FOR X=0 TO 6:FOR Y=0 TO 6:IF Y<X
THEN NEXT Y
8050 X(X,Y)=10-Y+2*X:Y(X,Y)=Y+2:NEXT Y
:NEXT X:T=0:FOR Y=0 TO 6:FOR X=0 TO 6:
IF X>Y THEN NEXT Y
8060 P(X,Y)=T:SX(T)=X:SY(T)=Y:T=T+1:NE
XT X:NEXT Y

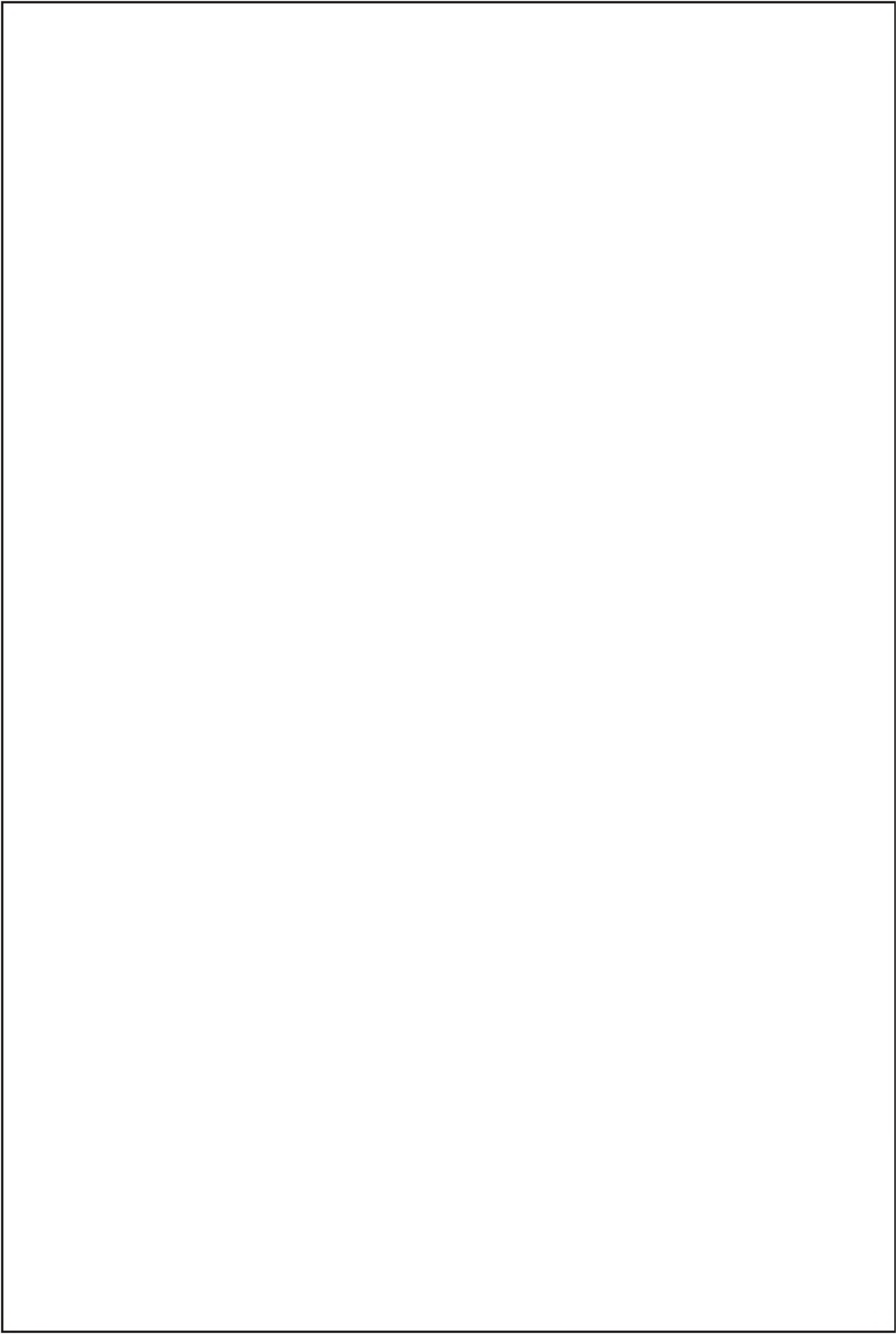
```

```

8100 FOR T=0 TO 26:READ K:B(T)=K:NEXT
T
8110 DATA 1,55,107,157,205,251,295,337
,377,415,451,485,517,547,575,601,625,6
47,667,685,701,715,727,737,745,751,755
8120 OPEN #1,4,0,"K:":T=PEEK(560)+256*
PEEK(561):FOR J=T+7 TO T+19 STEP 2:POK
E J,7:POKE J+1,112:NEXT J
8130 POKE T+22,65:POKE T+23,PEEK(560):
POKE T+24,PEEK(561):POSITION 5,5: ? #6;
"PLEASE WAIT"
8140 FOR T=1 TO 4:READ K:F$(T,T)=CHR$(
K):NEXT T:J$="/":DATA 171,44,13,142
8500 FOR T=0 TO 39:READ K:POKE 31744+8
8+T,K:NEXT T
8510 DATA 0,0,60,126,24,126,60,0
8520 DATA 0,0,126,195,189,195,126,0
8530 DATA 0,0,102,126,24,126,102,0
8540 DATA 0,0,24,126,231,126,24,0
8550 DATA 0,0,24,60,126,60,24,0
8600 C=0:FOR T=1 TO 756:READ K:C=C+K:A
$(T,T)=CHR$(K):NEXT T:RETURN
9000 DATA 2,32,1,32,5,32,32,32,3,32,9,
32,32,32,32,32,6,32,14,32
9010 DATA 32,32,32,32,32,32,10,32,20,3
2,32,32,32,32,32,32,32,15,32
9020 DATA 27,32,32,32,32,32,32,32,32,3
2,32,32,21,32,0,4,4,32,2,3
9030 DATA 7,32,8,32,5,32,6,32,11,32,13
,32,9,32,32,32,10,32,16,32
9040 DATA 19,32,14,32,32,32,32,15,3
2,22,32,26,32,20,32,32,32,32
9050 DATA 32,32,21,32,32,32,8,32,1,5,4
,32,13,32,9,32,3,32,7,32
9060 DATA 19,32,14,32,32,32,6,32,11,32
,26,32,20,32,32,32,32,10,32
9070 DATA 16,32,32,32,27,32,32,32,32,3
2,32,32,15,32,22,32,1,7,0,12
9080 DATA 7,32,4,6,2,11,17,32,12,32,8,
32,5,10,16,32,23,32,18,32
9090 DATA 13,32,9,32,15,32,22,32,32,32
,25,32,19,32,14,32,32,32,21,32
9100 DATA 32,32,32,32,2,8,12,32,3,8,5,
7,12,32,18,32,13,32,6,9
9110 DATA 11,32,17,32,25,32,19,32,14,3
2,10,32,16,32,23,32,32,26,32
9120 DATA 20,32,32,32,15,32,22,32,32,3
2,18,32,1,13,4,9,8,32,25,32
9130 DATA 19,32,3,14,7,32,12,32,32,32,
26,32,20,32,6,32,11,32,17,32
9140 DATA 32,32,32,32,27,32,32,32,10,3
2,16,32,23,32,3,11,1,17,0,24

```

9150 DATA 11, 32, 7, 10, 4, 16, 2, 23, 32, 32, 1  
 7, 32, 12, 32, 8, 15, 5, 22, 32, 32  
 9160 DATA 32, 32, 24, 32, 18, 32, 13, 32, 9, 21  
 , 32, 32, 32, 32, 32, 32, 4, 12, 2, 18  
 9170 DATA 17, 32, 6, 12, 8, 11, 5, 17, 24, 32, 2  
 4, 32, 18, 32, 10, 13, 9, 16, 23, 32  
 9180 DATA 32, 32, 32, 32, 25, 32, 19, 32, 14, 1  
 5, 22, 32, 32, 32, 32, 32, 5, 13, 24, 32  
 9190 DATA 3, 18, 7, 13, 9, 12, 18, 32, 32, 32, 2  
 5, 32, 6, 19, 11, 14, 17, 32, 24, 32  
 9200 DATA 32, 32, 32, 32, 26, 32, 10, 20, 16, 3  
 2, 23, 32, 32, 32, 32, 32, 1, 25, 4, 19  
 9210 DATA 8, 14, 13, 32, 32, 32, 32, 32, 3, 26,  
 7, 20, 12, 32, 18, 32, 32, 32, 32  
 9220 DATA 32, 32, 6, 27, 11, 32, 17, 32, 24, 32  
 , 6, 16, 3, 23, 1, 32, 0, 32, 16, 32  
 9230 DATA 11, 15, 7, 22, 4, 32, 2, 32, 32, 32, 2  
 3, 32, 15, 16, 12, 21, 8, 32, 5, 32  
 9240 DATA 32, 32, 32, 32, 7, 17, 4, 24, 2, 32, 2  
 3, 32, 10, 17, 12, 16, 8, 23, 5, 32  
 9250 DATA 32, 32, 32, 32, 24, 32, 15, 18, 13, 2  
 2, 9, 32, 32, 32, 32, 32, 8, 18, 5, 25  
 9260 DATA 32, 32, 6, 24, 11, 18, 13, 17, 9, 24,  
 32, 32, 32, 32, 32, 32, 10, 25, 16, 19  
 9270 DATA 14, 23, 32, 32, 32, 32, 9, 19, 32, 32  
 , 3, 32, 7, 25, 12, 19, 14, 18, 25, 32  
 9280 DATA 32, 32, 32, 32, 32, 6, 32, 11, 26, 17, 20  
 , 24, 32, 32, 32, 32, 32, 1, 32, 4, 32  
 9290 DATA 8, 26, 13, 20, 19, 32, 32, 32, 32, 32  
 , 3, 32, 7, 32, 12, 27, 18, 32, 25, 32  
 9300 DATA 10, 22, 6, 32, 3, 32, 1, 32, 0, 32, 22  
 , 32, 16, 21, 11, 32, 7, 32, 4, 32  
 9310 DATA 2, 32, 32, 32, 11, 23, 7, 32, 4, 32, 2  
 , 32, 32, 32, 15, 23, 17, 22, 12, 32  
 9320 DATA 8, 32, 5, 32, 32, 32, 12, 24, 8, 32, 5  
 , 32, 32, 32, 10, 32, 16, 24, 18, 23  
 9330 DATA 13, 32, 9, 32, 32, 32, 13, 25, 9, 32,  
 32, 32, 6, 32, 11, 32, 17, 25, 19, 24  
 9340 DATA 14, 32, 32, 32, 14, 26, 32, 32, 3, 32  
 , 7, 32, 12, 32, 18, 26, 20, 25, 32, 32  
 9350 DATA 32, 32, 1, 32, 4, 32, 8, 32, 13, 32, 1  
 9, 27, 26, 32, 15, 32, 10, 32, 6, 32  
 9360 DATA 3, 32, 1, 32, 0, 32, 16, 32, 11, 32, 7  
 , 32, 4, 32, 2, 32, 17, 32, 12, 32  
 9370 DATA 8, 32, 5, 32, 18, 32, 13, 32, 9, 32, 1  
 9, 32, 14, 32, 20, 32  
 9500 GRAPHICS 0:PRINT "INCORRECT DATA  
 IN LINES 9000-9370":END



# PART V

---

## Music and Noise



**A**tari BASIC has a powerful sound function built into it. The function, which takes four variables, is:

### SOUND A, B, C, D

The first variable, A, controls the number of voices or sounds the computer can use. It ranges from 0 to 3, thus controlling four voices—0, 1, 2 and 3.

The second variable, B, controls the pitch or note that the computer plays. It ranges from 0 to 255, giving you 256 sounds to control. These sounds are sometimes closer together than the sounds you'll find on the piano keyboard. Here is a chart giving you piano/computer equivalents for several octaves:

N	Note	N	Note
26	D#	61	C
27	D	65	B
29	C#	69	A#
31	C	73	A
32	B	77	G#
34	A#	82	G
36	A	86	F#
38	G#	92	F
41	G	97	E
43	F#	103	D#
46	F	109	D
48	E	115	C#
51	D#	122	C
54	D	129	B
58	C#	137	A#
		145	A

Notice that *the lower the number the higher the note!!!!*

The third variable, C, is a special effects variable. Sometimes it causes the equivalent of explosions or spooky sounds. When it is set at 10 you are in the normal music mode, which ranges from 0 to 15. Experiment with settings other than 10. Some will produce no sound

at all; other will create effects you might want to include in programs you create.

The fourth variable, D, controls the volume. It ranges from 0 to 15 with 8 being the normal mid-volume, 0 being the lowest setting, and 15 the loudest.

There is one other command that is used with SOUND A,B,C,D to give you control over the length of time each note is sounded. That is a pause command that is represented in BASIC by an empty FOR ... NEXT ... loop. For example:

```
FOR X=1 TO 100:NEXT X
```

The actual length of the pause created by the loop varies according to where it appears in your program, although, the larger the number that comes after TO the longer the pause. You'll get a feel for timing by playing around with pause loops.

Here are a few simple programs that will help you get a sense of how the SOUND command and the pause loop work together:

This program plays middle, high, and low C at the same time:

```
0 REM C IN 3 OCTAVES
10 SOUND 0,31,10,8
20 SOUND 1,61,10,8
30 SOUND 2,122,10,8
40 FOR X=1 TO 1500:NEXT X
```

This program also plays the three C's but starts with one, adds a second and then a third after slight pauses. Notice where the pauses are placed and how long they are when you run the program.

```
0 REM C'S IN THREE TIME ZONES
10 SOUND 0,31,10,8
15 FOR X=1 TO 500:NEXT X
20 SOUND 1,61,10,8
25 FOR X=1 TO 1000:NEXT X
30 SOUND 2,122,10,8
35 FOR X=1 TO 1500:NEXT X
```

This program only uses one voice and thus plays the three C's one after the other without creating a chord.



```

0 REM C'S IN THREE STAFFS
10 SOUND 0,31,10,8
15 FOR X=1 TO 500:NEXT X
20 SOUND 0,61,10,8
25 FOR X=1 TO 500:NEXT X
30 SOUND 0,122,10,8
35 FOR X=1 TO 500:NEXT X

```

Here the third or special effects variable is changed so that you can hear some of the sounds and noises you can make with your Atari 800.

```

0 REM SOUND AND EFFECTS
10 SOUND 0,31,10,8
15 FOR X=1 TO 500:NEXT X
20 SOUND 0,61,8,8
25 FOR X=1 TO 500:NEXT X
30 SOUND 0,122,6,8
35 FOR X=1 TO 500:NEXT X
0 REM SOUND, EFFECTS AND VOLUME
10 SOUND 0,31,10,1
15 FOR X=1 TO 500:NEXT X
20 SOUND 0,61,8,6
25 FOR X=1 TO 500:NEXT X
30 SOUND 0,122,6,12
35 FOR X=1 TO 500:NEXT X

```

This program takes you through all of the sounds you can control over two octaves. You'll notice that these sounds do not make up a scale as the computer can generate sounds that lie in between notes on a 12-tone scale.

```

0 REM COMPUTER RANGE 2 OCTAVES
10 FOR X=36 TO 145
20 SOUND 0,X,10,8
30 FOR Y=1 TO 50:NEXT Y
40 NEXT X

```

This final program shows you how to put sound into a READ DATA command. This is a memory-efficient way of storing tunes.

Notice that all the notes are put on line 50. You don't have to write a separate SOUND command for each note. Also notice that the -1 is used to let the computer know (at line 15) when the data has all been used.

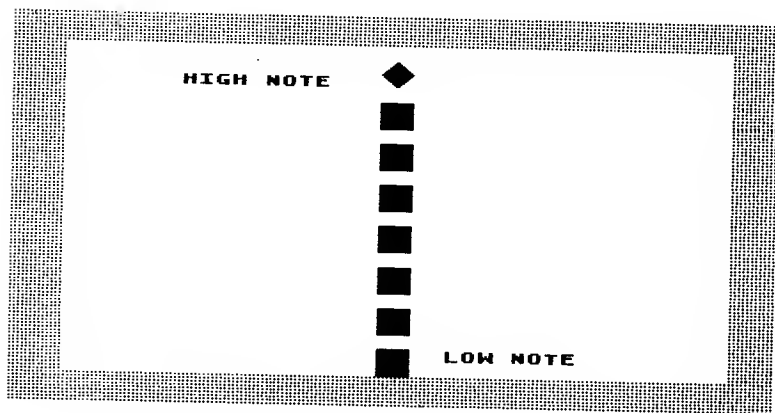
```
0 REM SOUND WITH FLAG:READ..DATA..
10 READ Z
15 IF Z=-1 THEN GOTO 45
20 SOUND 0,Z,10,8
30 FOR X=1 TO 50:NEXT X
40 GOTO 10
45 PRINT "THAT'S ALL FOLKS"
50 DATA 10,200,25,56,78,98,45,92,-1
```

Now here are several more complex programs that you might find fun and useful.

# 24

## High Note Low Note

In this game you use a joystick to rearrange a series of notes from high to low. Eight notes are scrambled and hidden under squares on the screen.



By pushing the joystick forward or backward, you move up and down the musical scale. Press the button to pick up a note in position and carry it to another position. Press the button again to drop the note. Put the lowest note at the bottom of the screen and the highest at the top. The computer will let you know when you have succeeded in ordering the notes.

```
100 REM *** HIGH NOTE LOW NOTE ***
110 DIM M$(8), A(7), P(7), B$(7), C$(7), D$(7):GRAPHICS 0:POKE 752,1: ?
120 FOR T=1 TO 7:READ K: B$(T,T)=CHR$(K)
130 NEXT T:DATA 160,160,30,30,29,160,160
140 FOR T=1 TO 7:READ K: C$(T,T)=CHR$(K)
150 NEXT T:DATA 6,7,30,30,29,7,6
160 FOR T=1 TO 7:READ K: D$(T,T)=CHR$(K)
170 NEXT T:DATA 8,10,30,30,29,138,136
```

```

200 PRINT CHR$(125):M$="*****":FOR
T=0 TO 7
210 R=INT(RND(1)*8):IF M$(R+1,R+1)<>"*
" THEN 210
220 M$(R+1,R+1)=STR$(T):P(R)=T:NEXT T
230 R=INT(RND(1)*4):RESTORE 900+R:FOR
T=0 TO 7:READ K:A(T)=K:NEXT T
240 POSITION 7,2:? "HIGH NOTE":FOR T=0
TO 7:POSITION 19,1+T*3:? B$;:S=A(P(T)
):GOSUB 500:NEXT T:POSITION 23,22
250 ? "LOW NOTE":Y=0:W=0:GOSUB 600
300 IF STRIG(0)=0 THEN 400
305 IF W=1 AND PEEK(53279)=6 THEN 200
310 IF STICK(0)=15 THEN 300
320 KY=Y:Z=STICK(0):IF Z=14 THEN Y=Y-1
:IF Y<0 THEN Y=0
330 IF Z=13 THEN Y=Y+1:IF Y>7 THEN Y=7
340 IF Y=KY THEN 300
350 POSITION 19,1+KY*3:? B$;:GOSUB 600
:GOTO 300
400 NN=1-NN:KY=Y:GOSUB 600
410 IF STRIG(0)=0 THEN 410
420 GOTO 300
500 FOR L=15 TO 0 STEP -1:SOUND 0,S,10
,L:NEXT L:SOUND 0,0,0:RETURN
600 POSITION 19,1+Y*3:IF NN THEN ? C$;
:T=P(Y):P(Y)=P(KY):P(KY)=T:M$(Y+1,Y+1)
=STR$(P(Y)):M$(KY+1,KY+1)=STR$(P(KY))
610 IF NN=0 THEN ? D$;
620 S=A(P(Y)):GOSUB 500:IF M$="0123456
7" AND NN=0 THEN 700
630 IF W=1 THEN 710
640 RETURN
700 POSITION 4,10:? "YOU DID IT!":POSI
TION 25,10:? "PRESS START":W=1:RETURN
710 POSITION 4,10:? " ":POSI
TION 25,10:? " ":W=0:RETURN
900 DATA 60,64,72,81,91,96,108,121
901 DATA 72,81,91,96,108,121,128,144
902 DATA 81,85,96,108,121,128,144,162
903 DATA 96,108,121,128,144,162,173,19
3

```

Here are the DATA statements used in the program. Notice that each line has eight numbers representing the eight notes used in the game and also that these numbers are ordered from low to high. You can change these statements and put your own scales or intervals in their place. Be sure, however, to keep eight notes on a line and order them from lowest to highest.

```

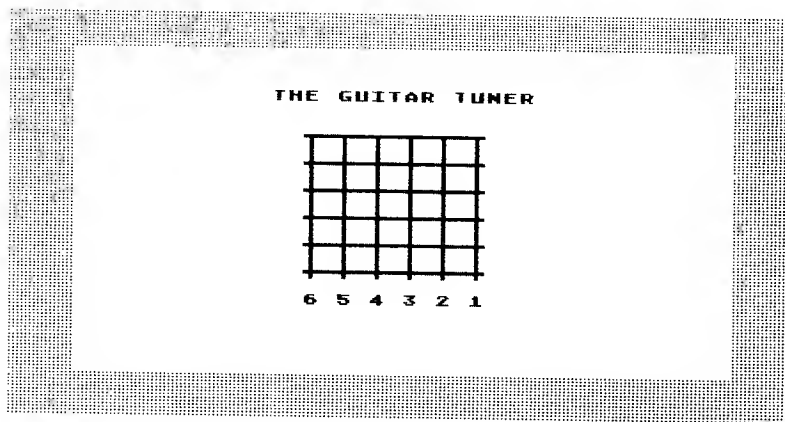
900 DATA 60,64,72,81,91,96,108,121
901 DATA 72,81,91,96,108,121,128,144
902 DATA 81,85,96,108,121,128,144,162
903 DATA 96,108,121,128,144,162,173,19
3

```

# 25

## The Guitar Tuner

You can use this program to tune your guitar. The screen presents you with a picture of a grid indicating the six strings on a guitar. Just press the number corresponding to any string to turn on that tone. When you want to turn off the tone, press any key.



```

100 REM   *** THE GUITAR TUNER ***
110 DIM S(5),A$(11),B$(11),C$(11):FOR
T=0 TO 5:READ K:S(T)=K:NEXT T:DATA 47,
64,81,108,144,193
120 FOR T=1 TO 11:READ K:A$(T,T)=CHR$(
K):READ K:B$(T,T)=CHR$(K):READ K:C$(T,
T)=CHR$(K):NEXT T
130 DATA 23,124,19,18,32,18,23,124,19,
18,32,18,23,124,19,18,32,18,23,124,19,

```

```

18,32,18,23,124,19,18,32,18,23,124,19
140 POKE 752,1:PRINT CHR$(125):POSITIO
N 12,3:PRINT "THE GUITAR TUNER":X=14:Y
=6
150 POSITION X,Y:PRINT A$:FOR T=1 TO 1
0 STEP 2:POSITION X,Y+T:PRINT B$:POSIT
ION X,Y+T+1:PRINT C$:NEXT T
160 POSITION X,Y+12:PRINT "6 5 4 3 2 1
":OPEN #1,4,0,"K:":POKE 764,255
170 IF PEEK(764)=255 THEN 170
180 GET #1,K:K=K-49:IF K<0 OR K>5 THEN
GOSUB 300:SOUND 0,0,0,0:GOTO 170
190 GOSUB 300:POSITION X+10-K*2,Y-1:PR
INT CHR$(20):SOUND 0,S(K),10,10:GOTO 1
70
300 POSITION X,Y-1:PRINT "
":RETURN

```

# 26

## Guitar Chord Tutor

Once you've got your guitar tuned up, you might want a little help with mastering guitar chords. This program will give you the fingering, name, and sound of the chords most commonly used on the guitar.

**GUITAR CHORD TUTOR**

C	G7	F
G	D7	C
D	A7	G
A	E7	D
E	B7	A

<NUMBER> Play string  
 <LETTER> Major chord  
 <CONTROL LETTER> Seventh chord

The Guitar Tutor program displays the top frets of a guitar finger board and the most used chords in five keys. Type the letter corresponding to the chord you want to learn. Press CONTROL plus a letter to select a major seventh chord. Press the numbers one through six to hear each note of the chord. The numbers on the frets tell you which fingers to use in forming the chord.

```

100 REM  ** GUITAR CHORD TUTOR **
110 DIM S(5,3),C(5),A$(11),B$(11),C$(1
1),T$(2):GRAPHICS 0:POKE 752,1:?
120 FOR T=0 TO 5:FOR J=0 TO 3:READ K
130 S(T,J)=K:NEXT J:NEXT T:DATA 47,45,
42,40,64,60,57,53,81,76,72,68,108,102,
96,91,144,136,128,121,193,182,173,162
140 FOR T=1 TO 11:READ K:A$(T,T)=CHR$(
K):READ K:B$(T,T)=CHR$(K):READ K:C$(T,
T)=CHR$(K):NEXT T:OPEN #1,4,0,"K:"
150 DATA 23,124,19,18,32,18,23,124,19,
18,32,18,23,124,19,18,32,18,23,124,19,
18,32,18,23,124,19,18,32,18,23,124,19
160 POSITION 10,1:? "GUITAR CHORD TUTO
R":X=21:Y=4:FOR T=0 TO 4:FOR J=0 TO 2:
READ T$:POSITION 5+J*4,5+T*2
170 ? T$:NEXT J:NEXT T:DATA C,G7,F,G,D
7,C,D,A7,G,A,E7,D,E,B7,A
200 POSITION 2,20:? "<NUMBER> Play str
ing":? "<LETTER> Major chord":? "<CONT
ROL LETTER> Seventh chord"
210 K=67:GOTO 400
300 GOSUB 900:RESTORE 800+FF+K:FOR T=0
TO 5:READ P:C(T)=P:READ T$
310 P=(P<>-1)*P:POSITION X+10-T*2,Y-1+
P*2:? T$:NEXT T
320 GET #1,K:IF K<49 OR K>54 THEN 400
330 K=K-49:GOSUB 950:POSITION X+10-K*2
,Y+14:? CHR$(20)
340 IF C(K)<>-1 THEN FOR T=15 TO 0 STE
P -1:SOUND 0,S(K,C(K)),10,T:FOR J=1 TO
2:NEXT J:NEXT T:SOUND 0,0,0,0
350 GOTO 320
400 IF K>0 AND K<8 THEN 450
410 IF K<65 OR K>71 THEN 320
420 K=K-65:IF K=1 THEN 320
430 POSITION 35,7:? CHR$(65+K);" ":FF=
0:GOTO 300
450 K=K-1:IF K=2 OR K=5 THEN 320
460 POSITION 35,7:? CHR$(65+K);"7":FF=
10:GOTO 300
800 DATA 0,0,2,4,2,3,2,2,0,0,0,0
802 DATA 0,0,1,1,0,0,2,2,3,3,-1,X

```

```

803 DATA 2,2,3,3,2,1,0,0,0,0,-1,X
804 DATA 0,0,0,0,1,1,2,3,2,2,0,0
805 DATA 1,1,1,1,2,2,3,4,3,3,1,1
806 DATA 3,3,0,0,0,0,0,0,2,1,3,2
810 DATA 0,0,2,3,0,0,2,2,0,0,0,0
811 DATA 2,4,0,0,2,3,1,1,2,2,-1,X
813 DATA 2,3,1,1,2,2,0,0,0,0,-1,X
814 DATA 0,0,3,4,1,1,2,3,2,2,0,0
816 DATA 1,1,0,0,0,0,0,0,2,2,3,3
900 POSITION X,Y-1:?" ":POS
ITION X,Y:?" A*:FOR T=1 TO 10 STEP 2:PO
SITION X,Y+T:?" B*
910 POSITION X,Y+T+1:?" C*:NEXT T:POSIT
ION X,Y+12:?" "6 5 4 3 2 1"
950 POSITION X,Y+14:?" " ":RE
TURN

```



# 27, 28

## Music Match and Noise Match

Here are two matching games involving sound. In the Music Match game you are presented with a grid with the letters from A to P printed in order in the squares. When you press a letter, you will hear a tune. Each tune is repeated twice on the grid. The goal of this two-player game is to press two letters on your turn that play the same tune. The game provides good ear training. You have to remember tunes and match them up. Each player presses two letters per turn. Whenever someone makes a match, however, they get another chance. Here's a picture of the grid followed by the game program:



\*\*\* THE MUSIC MATCH GAME \*\*\*

SCORE: 0

A	B	C	D
E	*	G	H
I	J	K	L
M	N	O	P

SCORE: 0

PLAYER 2

```

100 REM      *** MUSIC MATCH ***
110 GOTO 400
200 RESTORE 300+R*10
210 READ P:IF P=-1 THEN SOUND 0,0,0,0:
RETURN
220 READ D:SOUND 0,0,0,0:SOUND 0,P,10,
10:FOR T=1 TO D*40:NEXT T:GOTO 210
300 DATA 81,2,60,2,81,2,72,2,81,1,81,1
,60,2,81,2,72,2,81,1,81,1,60,2,81,2,72
,2,81,1,81,1,96,2,108,2,121,3,-1
310 DATA 108,2,81,2,81,2,81,2,108,2,96
,2,96,2,108,2,108,2,81,2,81,2,72,2,72,
2,64,4,81,4,-1
320 DATA 81,1,81,1,81,2,91,2,96,2,96,2
,108,2,96,2,91,4,108,4,-1
330 DATA 108,4,108,4,108,3,96,1,85,4,8
5,3,96,1,85,3,81,1,72,8,-1
340 DATA 108,2,64,1,64,1,64,1,64,1,60,
2,64,2,72,2,72,4,108,1,108,1,72,1,72,1
,72,1,72,1,64,1,64,1,72,2,81,2,81,4,-1
350 DATA 64,2,81,4,64,2,81,3,81,1,96,1
,108,3,81,1,81,1,81,1,81,1,64,1,53,1,5
3,2,53,4,-1
360 DATA 81,2,64,2,60,2,53,4,64,4,64,2
,81,2,64,2,60,2,53,4,64,2,47,2,53,4,64
,4,72,4,-1
370 DATA 81,4,81,3,72,1,64,4,64,4,72,3
,81,1,72,3,64,1,81,4,108,4,81,4,81,3,7
2,1,64,4,64,4,72,3,81,1,72,3,64,1
375 DATA 81,4,-1
400 DIM A$(16),G$(36),D$(1),X(15),Y(15
),K(1),R(1),S(1):GRAPHICS 0:POKE 752,1
:? :OPEN #1,4,0,"K:"

```

```

410 RESTORE 900:FOR T=1 TO 36:READ K:G
$(T,T)=CHR$(K):NEXT T
420 L=0:FOR Y=0 TO 3:FOR X=0 TO 3:X(L)
=16+X*2:Y(L)=7+Y*2:L=L+1:NEXT X:NEXT Y
500 ? CHR$(125):POSITION 6,2:? "*** TH
E MUSIC MATCH GAME ***":POKE 82,15
510 POSITION 15,6:? G$(1,9):? G$(10,18
):FOR T=1 TO 3:? G$(19,27):? G$(10,18)
:NEXT T:? G$(28,36):POKE 82,2
520 FOR T=0 TO 15:POSITION X(T),Y(T):?
CHR$(65+T):NEXT T:A$="*****
*":FOR T=0 TO 7:FOR J=0 TO 1
530 R=INT(RND(1)*16)+1:IF A$(R,R)<>"*"
THEN 530
540 A$(R,R)=STR$(T):NEXT J:NEXT T:MM=0
:S(0)=0:S(1)=0:PL=0:GOSUB 800
600 POSITION 4,12:? " ":IF PL=0
THEN POSITION 4,12:? "PLAYER 1"
610 POSITION 27,12:? " ":IF PL=
1 THEN POSITION 27,12:? "PLAYER 2"
620 FOR C=0 TO 1
630 D$=STR$(0):POKE 702,64:GET #1,K:K=
K-65:IF K<0 OR K>15 THEN 630
640 IF C=1 THEN IF K=K(0) THEN 630
650 IF A$(K+1,K+1)="*" THEN 630
660 POSITION X(K),Y(K):? "":R=VAL(A$(
K+1,K+1)):GOSUB 200:K(C)=K:R(C)=R:NEXT
C:IF R(0)=R(1) THEN 700
670 FOR C=0 TO 1:POSITION X(K(C)),Y(K(
C)):PRINT CHR$(65+K(C)):NEXT C:PL=1-PL
:GOTO 600
700 FOR C=0 TO 1:POSITION X(K(C)),Y(K(
C)):PRINT CHR$(160):A$(K(C)+1,K(C)+1)=
"*":NEXT C:S(PL)=S(PL)+10:GOSUB 800
710 MM=MM+1:IF MM<8 THEN 600
720 POSITION 2,17:IF S(0)=S(1) THEN ?
"TIE GAME!":GOTO 750
730 T=1:IF S(1)>S(0) THEN T=2
740 ? "YOU DID IT PLAYER ";T
750 POSITION 12,21:? "PRESS ANY KEY...
":POKE 764,255
760 IF PEEK(764)=255 THEN 760
770 POKE 764,255:GOTO 500
800 POSITION 4,8:? "SCORE: ";S(0):POS
ITION 27,8:? "SCORE: ";S(1):RETURN
900 DATA 17,18,23,18,23,18,23,18,5,124
,32,124,32,124,32,124,32,124,1,18,19,1
8,19,18,19,18,4,26,18,24,18,24,18,24
910 DATA 18,3

```

Here is the section of the program that contains the data for the tunes used in the game:

```

200 RESTORE 300+R*10
210 READ P: IF P=-1 THEN SOUND 0,0,0,0:
RETURN
220 READ D: SOUND 0,0,0,0: SOUND 0,P,10,
10: FOR T=1 TO D*40: NEXT T: GOTO 210
300 DATA 81,2,60,2,81,2,72,2,81,1,81,1
,60,2,81,2,72,2,81,1,81,1,60,2,81,2,72
,2,81,1,81,1,96,2,108,2,121,3,-1
310 DATA 108,2,81,2,81,2,81,2,108,2,96
,2,96,2,108,2,108,2,81,2,81,2,72,2,72,
2,64,4,81,4,-1
320 DATA 81,1,81,1,81,2,91,2,96,2,96,2
,108,2,96,2,91,4,108,4,-1
330 DATA 108,4,108,4,108,3,96,1,85,4,8
5,3,96,1,85,3,81,1,72,8,-1
340 DATA 108,2,64,1,64,1,64,1,60,
2,64,2,72,2,72,4,108,1,108,1,72,1,72,1
,72,1,72,1,64,1,64,1,72,2,81,2,81,4,-1
350 DATA 64,2,81,4,64,2,81,3,81,1,96,1
,108,3,81,1,81,1,81,1,81,1,64,1,53,1,5
3,2,53,4,-1
360 DATA 81,2,64,2,60,2,53,4,64,4,64,2
,81,2,64,2,60,2,53,4,64,2,47,2,53,4,64
,4,72,4,-1
370 DATA 81,4,81,3,72,1,64,4,64,4,72,3
,81,1,72,3,64,1,81,4,108,4,81,4,81,3,7
2,1,64,4,64,4,72,3,81,1,72,3,64,1
375 DATA 81,4,-1

```

Notice that, at line 210, the computer is instructed to READ P and also note that IF P=-1 THEN SOUND 0,0,0,0 which shuts the sound off. The -1 is a flag to indicate that a tune has been finished. It is used in the same way here as it was used in the short program in the introduction to this part. Also note that lines 300 to 375 contain the tunes, each one ending with a -1. If you want to create tunes for your own version of the game, simply replace these data statements with the numbers corresponding to the notes in your tunes.

The Noisy Match or, as we've called it, Noisy Concentration uses the same grid as Music Match and is played in the same way. However, it uses the special effects sounds that you can make with the SOUND function of the Atari 800. It is an easier game to play than Music Match and you might want to begin with it rather than with Music Match.

```

100 REM *** NOISY CONCENTRATION ***
110 GOTO 400
200 GOSUB 300+R*10: SOUND 0,0,0,0: SOUND
   1,0,0,0: RETURN
300 FOR T=1 TO 3: FOR J=20 TO 0 STEP -1
   : SOUND 0,J,10,10: NEXT J: NEXT T: RETURN
310 RESTORE 310: FOR J=1 TO 6: READ L: FO
   R T=1 TO 8: SOUND 0,L,10,10: SOUND 0,L+2
   ,10,10: NEXT T: NEXT J: RETURN
315 DATA 12,6,7,11,5,9
320 FOR T=0 TO 100 STEP 2: SOUND 0,T,10
   ,10: SOUND 1,100-T,10,10: NEXT T: RETURN
330 FOR T=12 TO 5 STEP -1: FOR L=15 TO
   0 STEP -2: SOUND 0,T,10,L: NEXT L: NEXT T
   : RETURN
340 FOR T=1 TO 9: SOUND 0,T*T,10,10: FOR
   J=T TO T+10: SOUND 1,J,10,10: NEXT J: NE
   XT T: RETURN
350 FOR T=1 TO 40: SOUND 0,5,4,10: SOUND
   1,5,4,10: NEXT T: RETURN
360 FOR T=1 TO 25: FOR J=0 TO 1: SOUND 0
   ,252+J,10,10: SOUND 1,254+J,10,10: NEXT
   J: NEXT T: RETURN
370 FOR T=0 TO 18 STEP 2: SOUND 0,T,2,1
   5: FOR J=1 TO 30: NEXT J: NEXT T: RETURN
400 DIM A$(16),G$(36),D$(1),X(15),Y(15
   ),K(1),R(1),S(1): GRAPHICS 0: POKE 752,1
   : ? : OPEN #1,4,0,"K:"
410 RESTORE 900: FOR T=1 TO 36: READ K: G
   $(T,T)=CHR$(K): NEXT T
420 L=0: FOR Y=0 TO 3: FOR X=0 TO 3: X(L)
   =16+X*2: Y(L)=7+Y*2: L=L+1: NEXT X: NEXT Y
500 ? CHR$(125): POSITION 6,2: ? "**** NO
   ISY CONCENTRATION ****": POKE 82,15
510 POSITION 15,6: ? G$(1,9): ? G$(10,18
   ): FOR T=1 TO 3: ? G$(19,27): ? G$(10,18)
   : NEXT T: ? G$(28,36): POKE 82,2
520 FOR T=0 TO 15: POSITION X(T),Y(T): ?
   CHR$(65+T): NEXT T: A$="*****
   *": FOR T=0 TO 7: FOR J=0 TO 1
530 R=INT(RND(1)*16)+1: IF A$(R,R)<>"*
   THEN 530
540 A$(R,R)=STR$(T): NEXT J: NEXT T: MM=0
   : S(0)=0: S(1)=0: PL=0: GOSUB 800
600 POSITION 4,12: ? " " " : IF PL=0
   THEN POSITION 4,12: ? "PLAYER 1"
610 POSITION 27,12: ? " " " : IF PL=
   1 THEN POSITION 27,12: ? "PLAYER 2"
620 FOR C=0 TO 1
630 D$=STR$(0): POKE 702,64: GET #1,K: K=
   K-65: IF K<0 OR K>15 THEN 630
640 IF C=1 THEN IF K=K(0) THEN 630
650 IF A$(K+1,K+1)="*" THEN 630

```

```

660 POSITION X(K),Y(K):? "*":R=VAL(A$(
K+1,K+1)):GOSUB 200:K(C)=K:R(C)=R:NEXT
C:IF R(0)=R(1) THEN 700
670 FOR C=0 TO 1:POSITION X(K(C)),Y(K(
C)):PRINT CHR$(65+K(C)):NEXT C:PL=1-PL
:GOTO 600
700 FOR C=0 TO 1:POSITION X(K(C)),Y(K(
C)):PRINT CHR$(160):A$(K(C)+1,K(C)+1)=
"*":NEXT C:S(PL)=S(PL)+10:GOSUB 800
710 MM=MM+1:IF MM<8 THEN 600
720 POSITION 2,17:IF S(0)=S(1) THEN ?
"TIE GAME!";:GOTO 750
730 T=1:IF S(1)>S(0) THEN T=2
740 ? "YOU DID IT PLAYER ";T
750 POSITION 12,21: ? "PRESS ANY KEY..."
:POKE 764,255
760 IF PEEK(764)=255 THEN 760
770 POKE 764,255:GOTO 500
800 POSITION 4,8: ? "SCORE: ";S(0):POSIT
ION 27,8: ? "SCORE: ";S(1):RETURN
900 DATA 17,18,23,18,23,18,23,18,5,124
,32,124,32,124,32,124,32,124,1,18,19,1
8,19,18,19,18,4,26,18,24,18,24,18,24
910 DATA 18,3

```

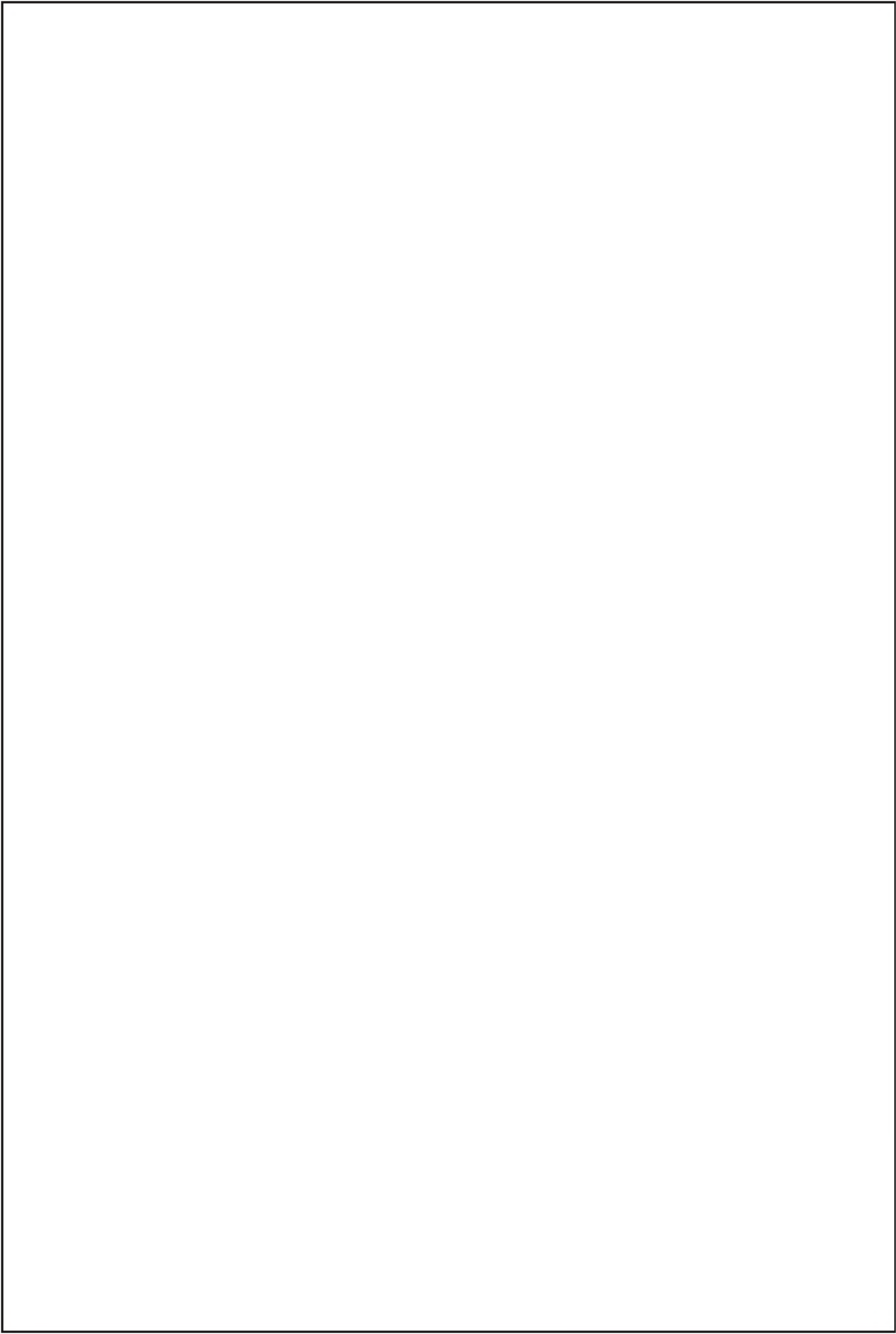
If you compare the programs for these two games you'll see that they are essentially identical except for the DATA statements on lines 200 to 375. Here are the data lines for Noisy Concentration. Notice that they change the B variable in the SOUND A,B,C,D command. You can fiddle around with them and create your own effects.

```

200 GOSUB 300+R*10:SOUND 0,0,0,0:SOUND
1,0,0,0:RETURN
300 FOR T=1 TO 3:FOR J=20 TO 0 STEP -1
:SOUND 0,J,10,10:NEXT J:NEXT T:RETURN
310 RESTORE 310:FOR J=1 TO 6:READ L:FO
R T=1 TO 8:SOUND 0,L,10,10:SOUND 0,L+2
,10,10:NEXT T:NEXT J:RETURN
315 DATA 12,6,7,11,5,9
320 FOR T=0 TO 100 STEP 2:SOUND 0,T,10
,10:SOUND 1,100-T,10,10:NEXT T:RETURN
330 FOR T=12 TO 5 STEP -1:FOR L=15 TO
0 STEP -2:SOUND 0,T,10,L:NEXT L:NEXT T
:RETURN
340 FOR T=1 TO 9:SOUND 0,T*T,10,10:FOR
J=T TO T+10:SOUND 1,J,10,10:NEXT J:NE
XT T:RETURN
350 FOR T=1 TO 40:SOUND 0,5,4,10:SOUND
1,5,4,10:NEXT T:RETURN

```

```
360 FOR T=1 TO 25:FOR J=0 TO 1:SOUND 0
,252+J,10,10:SOUND 1,254+J,10,10:NEXT
J:NEXT T:RETURN
370 FOR T=0 TO 18 STEP 2:SOUND 0,T,2,1
5:FOR J=1 TO 30:NEXT J:NEXT T:RETURN
```



# PART VI

---

## Computer Utilities





**A** *computer utility* is a program that can be used in many different contexts. A good example of a utility is a word processing program. Such programs can be used to write business or personal letters, take notes, or write a poem, novel, or science fiction book. It can also be used to lay out a newsletter or advertising brochure. You determine the use given the constraints of the program. Without your use, however, it does nothing. This is quite different than, say, a preprogrammed game that runs one way and can only be used as the programmer intended.

In this section we'll provide you with a number of simple utilities. For example, one may be the world's cheapest and shortest word processor. Others will allow you to tailor graphs for yourself, to set up a ballot and ballot counter, and turn your computer into a calculator. Some of the utilities will have wide applications and others will be more specialized. Once you begin to feel comfortable as a programmer you should think of creating a library of your own utilities and subroutines that is stored on disk. That way, you will be able to add them to other programs you build or use them without having to go through the tedious process of typing them in. It's a good idea to number your subroutines and utilities beginning somewhere around 2000. That way, you can add them onto other programs just by loading them on top of programs that have lower numbers.

## Scratch Pad: A Mini-Word Processor

This program provides a quick and easy 40-column word processor. When you run this program, you'll see nothing but a blank screen and the cursor. That's how it should be. There are no frills here, just a blank page for you to write on. Type anything you want. Use upper and lower case letters. All of the normal editing functions of the Atari are available to you, including clear screen, insert space, delete space, backspace, line insert and delete, and tab. When you near the end of a line, press **RETURN** just as you would on a typewriter. After you are done and want to print out what you have written, turn on your printer and press the **START** key. Whatever is on the screen will print out. To stop the printing before it's finished, press **START** again.

Here's a little shape poem created with the sketch pad and printed out on an EPSON MX-80 printer:

```

      OV          OV
    LOVE        LOVE
  LOVELOV   LOVELOVE
LOVELOVELOVELOVELOVEL
LOVELOVELOVELOVELOVELOV
LOVELOVELOVELOVELOVELOV
  LOVELOVELOVELOVELOV
    LOVELOVELOVELOV
      LOVELOVEL
        LOVELOV
          LOVEL
            LOV
              I

```

```

100 REM *** SCRATCH PAD ***
110 GRAPHICS 0:DIM T$(40)
120 OPEN #1,4,0,"K:":
200 IF PEEK(53279)="6 THEN 500
210 IF PEEK(764)=255 THEN 200
220 GET #1,K
230 IF K>27 AND K<96 THEN 400
240 IF K<96 AND K<123 THEN 400

```

```

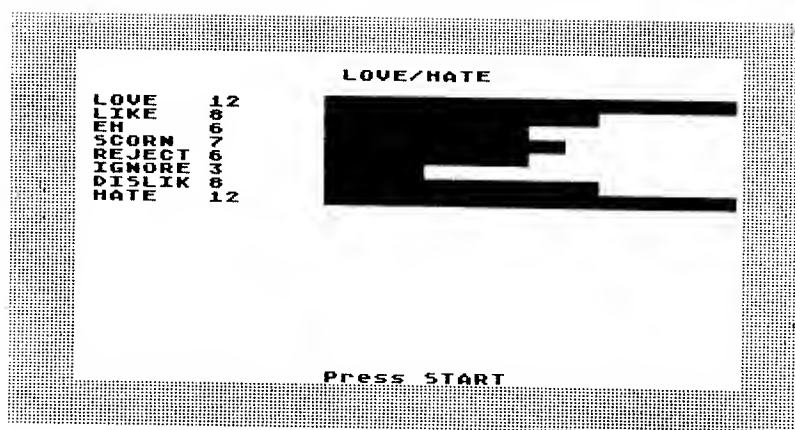
250 IF K>123 AND K<128 THEN 400
260 IF K>154 AND K<160 THEN 400
270 IF K=254 OR K=255 THEN 400
280 IF PEEK(694)<>0 THEN POKE 694,0
290 IF PEEK(702)>64 THEN POKE 702,64
300 GOTO 200
400 PRINT CHR$(K);
410 GOTO 200
500 TRAP 200
510 M=PEEK(88)+256*PEEK(89)
520 FOR Y=0 TO 23:B=M+Y*40
530 FOR X=0 TO 39:T=PEEK(B+X)
540 IF T<64 THEN T=T+32
550 T$(X+1,X+1)=CHR$(T):NEXT X
560 LPRINT T$
570 IF PEEK(53279)=6 THEN 600
580 NEXT Y:GOTO 200
600 IF PEEK(53279)=6 THEN 600
610 GOTO 200

```

# 30

## Graph Generator

This short program will produce a bar graph with up to 20 different bars. You can enter as many as 20 different labels and values. You can use up to seven-place numbers. The program will automatically scale the values you place in it so that what you will see is a graph of the relative sizes of the values of your labels. The values will also be printed out next to the labels as in this graph of the LOVE/HATE continuum:



In each graph, the largest number will fill up the entire space allotted to the graph. Here is the program, followed by some suggestions of the kinds of graphs you might want to experiment with:

```

100 REM *** GRAPH GENERATOR ***
110 DIM V(19),G(2),T$(30),L$(6),A$(120)
120 GRAPHICS 0:POKE 752,1:G(0)=22:G(1)=2
130 G(2)=130:POKE 82,1
200 PRINT CHR$(125):POSITION 10,1:PRINT "BAR GRAPH GENERATOR"
210 POSITION 3,3:PRINT "GRAPH TITLE: "
220 INPUT T$:POSITION 2,3:PRINT CHR$(156)
230 M=0:FOR T=0 TO 19:POSITION 3,T+3:PRINT "#";T+1:POSITION 7,T+3:PRINT "LABEL: "
240 POSITION 14,T+3:INPUT L$
250 TRAP 300:POSITION 30,T+3:INPUT V
260 IF V>M THEN M=V
270 A$(T*6+1,T*6+6)=L$:V(T)=V:NEXT T
300 TRAP 40000:L=T-1:PRINT CHR$(125):POSITION 20-LEN(T$)/2,1:PRINT T$
310 M=100/M:FOR T=0 TO L:POSITION 1,T+3:PRINT A$(T*6+1,T*6+6):POSITION 8,T+3:PRINT V(T)
500 K=INT(V(T)*M):HV=INT(K/4):LV=K-HV*4:POSITION 15,T+3
510 IF HV>0 THEN FOR J=1 TO HV:PRINT CHR$(160):NEXT J
520 IF LV>0 THEN PRINT CHR$(G(LV-1))

```

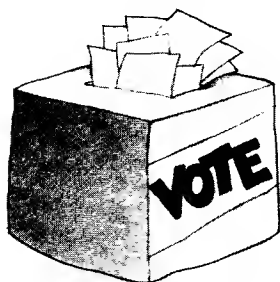
```

530 NEXT T:POSITION 15,23:PRINT "Press
    START";
600 IF PEEK(53279)<>6 THEN 600
610 POKE 764,255:GOTO 200

```

Some graphing ideas:

- Graph school grades.
- Graph your monthly expenses and see the relative proportions of your income that you spend on food, rent, entertainment, etc.
- Graph income distribution, age distribution, voting patterns, and other demographics of your community.
- Graph the won-lost percentages of your favorite teams and see which teams have done the best over a period of time.
- Do a personal preference test or a survey of people's favorite records or movies and see the relative positions of the first 20 printed out in graph form.

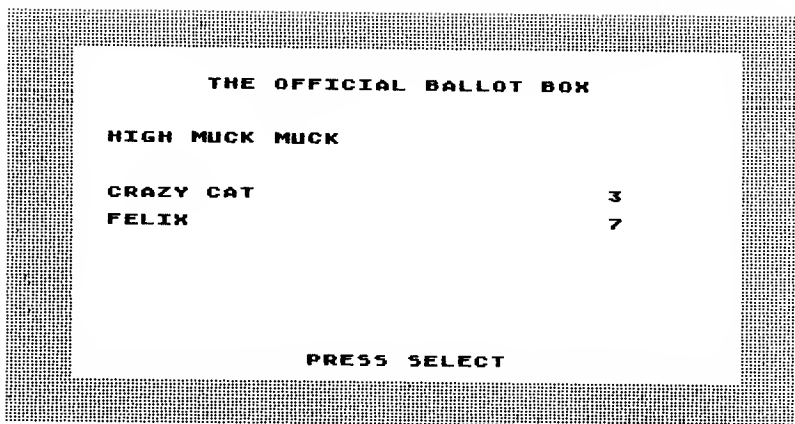


# 31

---

## The Ballot Box

This is a voting utility. It allows you to set up anywhere from 1 to 5 different offices or positions to vote for. In each position, you can have up to five candidates. After the ballot is set up you can use your computer as a voting machine or you can tally individual ballots with it yourself. You can save your ballot on disk, load it any time you want and have tallies of the results of the voting listed out for you as in this screen dump for the election of a HIGH MUCK MUCK:



This utility can be used for school or club elections, popularity polls, voter preference surveys, etc. . . . anything that requires a vote can be recorded and tallied using it.

```
100 REM *** THE BALLOT BOX ***
110 DIM A$(150),B$(750),T$(30),C(4),V(
4,4):OPEN #1,4,0,"K:":GOSUB 910
120 GRAPHICS 0:POKE 752,1:? :DATA 1. S
et up the ballot,2. Save ballot to dis
k,3. Load ballot from disk,4. Vote
200 GOSUB 900:RESTORE :FOR T=1 TO 4:RE
AD T$:? T$?:NEXT T
210 GET #1,K:K=K-48:IF K<1 OR K>4 THEN
210
220 ON K GOTO 300,500,550,600
300 GOSUB 910:GOSUB 900:? "HOW MANY OF
FICES? (1-5)":GET #1,P:P=P-49:IF P<0 O
R P>4 THEN 300
310 FOR M=0 TO P:GOSUB 900:? "ENTER NA
ME OF OFFICE #";M+1:? :INPUT T$:IF T$=
"" THEN M=M-1:NEXT M
320 A$(M*30+1,M*30+30)=T$
330 GOSUB 900:? T$:? :? "HOW MANY CAND
IDATES? (1-5)":GET #1,C:C=C-49:IF C<0
OR C>4 THEN 330
340 C(M)=C:FOR T=0 TO C:POSITION 2,10+
T*2:? CHR$(156):POSITION 2,10+T*2:? "C
ANDIDATE #";T+1;" ";:INPUT T$
350 IF T$="" THEN T=T-1:NEXT T
360 B$(M*150+T*30+1,M*150+T*30+30)=T$:
NEXT T:NEXT M:GOTO 200
```

```

500 TRAP 520:CLOSE #2:OPEN #2,8,0,"D:B
ALLOT.DAT":PUT #2,P:FOR T=0 TO P:PUT #
2,C(T):NEXT T
510 FOR M=0 TO P:? #2;A$(M*30+1,M*30+3
0):FOR T=0 TO C(M):? #2;B$(M*150+T*30+
1,M*150+T*30+30):NEXT T:NEXT M
520 CLOSE #2:GOTO 200
550 TRAP 520:CLOSE #2:OPEN #2,4,0,"D:B
ALLOT.DAT":GET #2,P:FOR T=0 TO P:GET #
2,K:C(T)=K:NEXT T
560 FOR M=0 TO P:INPUT #2;T$:A$(M*30+1
,M*30+30)=T$:FOR T=0 TO C(M):INPUT #2;
T$:B$(M*150+T*30+1,M*150+T*30+30)=T$
570 NEXT T:NEXT M:GOTO 520
600 FOR T=0 TO 4:FOR J=0 TO 4:V(T,J)=0
:NEXT J:NEXT T
610 GOSUB 900:? "PRESS START TO CAST Y
OUR BALLOT":? :? "PRESS OPTION TO END
THE VOTING"
620 IF PEEK(53279)=3 THEN 800
630 IF PEEK(53279)<>6 THEN 620
700 FOR M=0 TO P:GOSUB 900:? A$(M*30+1
,M*30+30):FOR T=0 TO C(M):POSITION 2,1
0+T*2
710 ? T+1;". ";B$(M*150+T*30+1,M*150+T
*30+30):NEXT T
720 GET #1,V:V=V-49:IF V<0 OR V>C(M) T
HEN 720
730 V(M,V)=V(M,V)+1:NEXT M:GOTO 610
800 FOR M=0 TO P:GOSUB 900:? A$(M*30+1
,M*30+30):FOR T=0 TO C(M)
810 POSITION 2,10+T*2:? B$(M*150+T*30+
1,M*150+T*30+30);V(M,T):NEXT T:POSITIO
N 14,22:? "PRESS SELECT"
820 IF PEEK(53279)<>5 THEN 820
830 NEXT M:GOTO 200
900 ? CHR$(125):POSITION 8,2:? "THE OF
FICIAL BALLOT BOX":POSITION 2,6:RETURN

910 B$(1)=" ":B$(750)=" ":B$(2)=B$:A$=
B$:RETURN

```

# 32

## Address Card File

The Address Card File can be used to store filing cards with any six different pieces of information. The form we are providing asks a person's name, address, city/state, home phone, and work phone; it also provides an extra line for remarks as in this entry:

*** ADDRESS CARD FILE ***	
Name	JULIUS CAESAR
Address	IMPERIAL PALACE
City/State	ROME, ITALY
Home Phone	ROYAL-22
Work Phone	NONE
Remarks	HAS TEMPER

1. Flip forward
2. Flip backward
3. Flip to letter
4. Add a new card
5. Delete current card
6. Modify current card
7. Load file from disk
8. Save file to disk

As you can see, the bottom of the screen shows you the options available to you with the program. You can flip forward and backward to see other cards, begin at any letter (the program automatically alphabetizes your entries), add a new entry, change an old entry, and save and retrieve your cards on disk. Here is the program:

```
100 REM *** ADDRESS CARD FILE ***
110 AZ=150:FZ=150:SZ=AZ*150:DIM N$(SZ)
    ,B$(FZ),K$(FZ),T$(25):OPEN #1,4,0,"K:"
    :GOSUB 810:B$=N$
120 C=AZ-1:P=C:GRAPHICS 0:POKE 752,1:P
    OKE 82,2:POSITION 7,2:? "*** ADDRESS C
    ARD FILE ***":POSITION 2,6
130 FOR T=1 TO 6:READ T$:FOR J=1 TO 10
    :? CHR$(128+ASC(T$(J,J))):NEXT J:PRIN
    T :NEXT T
```



```

140 POKE 82,10:POSITION 10,15:FOR T=1
TO 8:READ T$:? T$; ". ";T$:NEXT T:GOSUB
850
200 GET #1,K:K=K-48:IF K<1 OR K>8 THEN
200
210 POP :POP :ON K GOTO 300,350,400,45
0,500,550,600,650
300 P=P-1:IF P<C+1 THEN P=AZ-1
310 GOSUB 850:GOTO 200
350 P=P+1:IF P=AZ THEN P=C+1:IF P=AZ T
HEN P=AZ-1
360 GOSUB 850:GOTO 200
400 POSITION 2,4:? "WHICH LETTER?":GET
#1,K:K$=CHR$(K):IF K$>"a" AND K$<="z
" THEN K$=CHR$(K-32)
410 IF K$<"A" OR K$>"Z" THEN 440
420 FOR J=AZ-1 TO C+1 STEP -1:IF N$(J*
FZ+1,J*FZ+1)<K$ THEN NEXT J:GOTO 440
430 P=J:GOSUB 850
440 POSITION 0,4:? B$(1,39):GOTO 200
450 GOSUB 800
455 POKE 82,13:FOR T=0 TO 5:POSITION 1
3,6+T:POKE 752,0:INPUT T$:IF T>0 THEN
GOTO 470
460 FOR J=AZ-1 TO C STEP -1:IF T$>N$(J
*FZ+1,J*FZ+25) THEN NEXT J:J=J+1
465 N$(1,J*FZ)=N$(151,(J+1)*FZ):N$(J*F
Z+1,J*FZ+FZ)=B$
470 N$(J*FZ+T*25+1,J*FZ+T*25+25)=T$:NE
XT T:POKE 752,1:P=J:GOSUB 850:C=C-1:GO
TO 200
500 GOSUB 510:GOSUB 850:GOTO 200
510 IF C=AZ-1 THEN RETURN
520 FOR J=P TO C+1 STEP -1:K=J-1:N$(J*
FZ+1,J*FZ+FZ)=N$(K*FZ+1,K*FZ+FZ):NEXT
J
530 C=C+1:IF P=C THEN P=AZ-1
540 RETURN
550 IF C=AZ-1 THEN 200
560 GOSUB 520:GOTO 455
600 POSITION 3,4:? "REPLACE FILE IN ME
MORY? Y/N":GET #1,K:IF CHR$(K)="Y" OR
CHR$(K)="y" THEN 620
610 POSITION 0,4:? B$(1,39):CLOSE #2:G
OTO 200
620 TRAP 610:CLOSE #2:OPEN #2,4,0,"D:A
DDRESS.DAT"
630 GET #2,C:FOR J=AZ-1 TO C STEP -1:I
NPUT #2,K$:N$(J*FZ+1,J*FZ+FZ)=K$:NEXT
J:P=AZ-1:GOSUB 850:GOTO 610
650 POSITION 3,4:? "REPLACE FILE ON DI
SK? Y/N":GET #1,K:IF CHR$(K)<>"Y" AND

```

```

CHR$(K)<>"y" THEN 610
660 TRAP 610:CLOSE #2:OPEN #2,8,0,"D:ADDRESS.DAT":PUT #2,C
670 FOR J=AZ-1 TO C STEP -1:PRINT #2;N$(J*FZ+1,J*FZ+FZ):NEXT J:GOTO 610
800 FOR T=0 TO 5:POSITION 13,6+T:PRINT B$(1,26):NEXT T:RETURN
810 N$(1)=" ":N$(SZ)=" ":N$(2)=N$:RETURN
850 POKE 82,13:POSITION 13,6:FOR T=0 TO 5:? " ":N$(P*FZ+T*25+1,P*FZ+T*25+25):NEXT T:RETURN
900 DATA Name ,Address ,City/State,Home Phone,Work Phone,Remarks ,Flip forward,Flip backward
910 DATA Flip to letter,Add a new card,Delete current card,Modify current card,Load file from disk
920 DATA Save file to disk

```

Although the program is set up for addresses, it can easily be changed so that you can put recipes on it or organize a collection. For example, if you collect baseball cards, you can set up the program so that it stores:

PLAYERS NAME:  
 TEAM:  
 POSITION:  
 BATTING AVERAGE:  
 FIELDING PERCENT:  
 COMMENTS:

To do this, you have to change the data line 900. If you also want to organize your baseball cards alphabetically by team, put the TEAM category first and all of the players on the same team will be grouped together.

# 33

## A Calculator Utility

For all of the sophistication of home computers, it can become frustrating when you can't use them as simple calculators. In the future, calculator functions will probably be built into the machines. However, if you don't feel like waiting, here is a simple program you can copy and save to disk that will allow you to use your computer as a calculator whenever you care to. You can also add this program to other programs and build in a small calculator to games or other programs where fast calculation is useful.

To use this program, you enter values and operators (+-\*/=) as you would with any calculator. Press "C" for a complete clear. Press **ESC** to clear only the current entry. Press "S" to change the sign value of the value currently in the window.

```
100 REM *** THE CALCULATOR ***
110 DIM T$(16),K$(1),D$(1):OPEN #1,4,0
  ,K:":GRAPHICS 0:POKE 752,1
120 ? CHR$(125):POSITION 13,6:? "THE C
ALCULATOR":POSITION 10,8:? CHR$(17);
130 FOR T=1 TO 18:? CHR$(18);:NEXT T:?
  CHR$(5):POSITION 10,9:? CHR$(124):POS
ITION 29,9:? CHR$(124):POSITION 10,10
140 ? CHR$(26);:FOR T=1 TO 18:? CHR$(1
8);:NEXT T:? CHR$(3)
150 T$=""
200 D$=STR$(0):GET #1,K:K$=CHR$(K)
210 IF K$="+" THEN P=1:POSITION 34,9:P
RINT "+":GOTO 500
220 IF K$="-" THEN P=2:POSITION 34,9:P
RINT "-":GOTO 500
230 IF K$="*" THEN P=3:POSITION 34,9:P
RINT "*":GOTO 500
240 IF K$="/" THEN P=4:POSITION 34,9:P
RINT "/":GOTO 500
250 IF K$="=" THEN GOSUB 900:FF=1:AA=0
:GOTO 150
260 IF K$="C" THEN GOSUB 400:AA=0:KN=0
:N=0:P=0:POSITION 34,9:PRINT " ":GOTO
150
270 IF K$=CHR$(27) THEN GOSUB 400:N=0:
GOTO 150
```

```

280 IF K$="." THEN 300
290 IF K$="S" AND N<>0 THEN N=-N:T$=ST
R$(N):GOSUB 400:GOTO 340
295 IF K$<"0" OR K$>"9" THEN 200
300 IF LEN(T$)=16 THEN 200
310 T$(LEN(T$)+1)=K$
320 IF T$<> "." THEN N=VAL(T$)
330 IF FF=1 THEN GOSUB 400:FF=0
340 POSITION 28-LEN(T$),9:PRINT T$
350 GOTO 200
400 POSITION 12,9
410 PRINT "":RETURN
500 FF=1:IF AA=0 THEN AA=1:KN=N:KP=P:G
OTO 150
510 IF KP<>P THEN T=P:P=KP:KP=T:GOSUB
900:P=KP:GOTO 150
520 GOSUB 900:GOTO 150
900 ON P GOTO 910,920,930,940
910 N=N+KN:GOTO 950
920 N=KN-N:GOTO 950
930 N=N*KN:GOTO 950
940 IF N<>0 THEN N=KN/N:GOTO 950
950 T$=STR$(N):GOSUB 400:POSITION 28-L
EN(T$),9:PRINT T$:KN=N:RETURN

```

# 34

## Dungeon Dice

This utility allows you to choose the number of dice you want to roll and the number of sides per die. You can choose up to 8 dice and up to 20 sides per die. In addition, it calculates the percentage of the possible maximum score that your roll produces. For people who have played Dungeons and Dragons, the convenience of this program will probably be apparent. For anyone who wants to design a computer game with a complex element of choice built in, this can be a useful game subroutine.

Here is an example of what the program can do. Suppose you choose to roll five 10-sided dice. The computer will print out the results on each die and then tell you the percentage of the maximum

score that your roll achieved. (In this case the maximum is  $5 \times 10 = 50$ .)  
Here is a possible roll:

DIE	1	2	3	4	5
SCORE	5	7	3	9	1

PERCENTAGE =  $(\text{TOTAL} / 50) \times 100 = 25 / 50 \times 100 = 50\%$

```

100 REM *** DUNGEON DICE ***
110 DIM D$(26):DATA 17,18,18,5,29,30,3
0,30,30,124,32,32,124,29,30,30,30,30,2
6,18,18,3,28,30,30,30
120 GRAPHICS 0:POKE 752,1:? :FOR T=1 T
O 26:READ K:D$(T,T)=CHR$(K):NEXT T
200 GOSUB 500:POSITION 2,6:? "HOW MANY
DICE? (1-8) ";;INPUT N:IF N<1 OR N>8
THEN 200
210 POSITION 2,8:? CHR$(156):POSITION
2,8:? "HOW MANY SIDES? (4-20) ";;INPUT
S:IF S<4 OR S>20 THEN 210
220 P=S*N:N=N-1:GOSUB 500
230 POSITION 8,19:? "PRESS ANY KEY TO
ROLL":POSITION 6,21:? "PRESS START TO
CHANGE DICE":X=20-((N+1)*2)
300 POKE 764,255
310 IF PEEK(53279)=6 THEN 200
320 IF PEEK(764)=255 THEN 310
330 FOR J=0 TO INT(RND(1)*3)+2:C=0:FOR
T=0 TO N:POSITION X+T*4,7:R=INT(RND(1
)*S)+1:? D$;R
340 SOUND 0,15+J,10,8:SOUND 1,20+T,10,
6:C=C+R:NEXT T:NEXT J:SOUND 0,0,0,0:SO
UND 1,0,0,0
350 POSITION 15,13:? "TOTAL: ";C;" "
POSITION 12,14:? "POSSIBLE: ";P;" "
360 POSITION 10,15:? "PERCENTAGE: ";IN
T(C/P*100);"% " :GOTO 300
500 ? CHR$(125):POSITION 10,2:? "DUNGE
ON DICE ROLLER":RETURN

```

# 35

## Circle and Ellipse

Drawing a circle in BASIC is difficult and time-consuming. Drawing an ellipse is almost impossible. Here are four simple programs using the trigonometric SINE and COSINE functions that allow you to draw circles and ellipses in the Graphics 7 mode of your Atari. They also illustrate how you change the size of the circle and ellipse, which you can place anywhere on the screen. If you are drawing in BASIC or doing animation they are very useful routines to be able to call on:

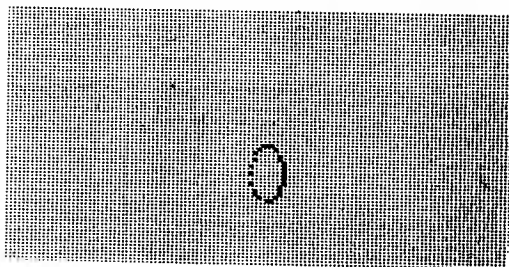
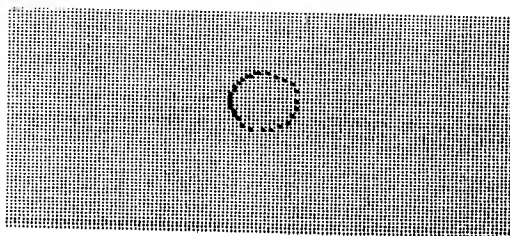
```
100 REM      ***  CIRCLE  ***
110 GRAPHICS 7:COLOR 1
120 FOR C=1 TO 22
130 A=A+0.28: X=SIN(A)*8: Y=COS(A)*8
140 PLOT X+80,Y+40
150 NEXT C
```

```
100 REM      ***  BIG CIRCLE  ***
110 GRAPHICS 7:COLOR 1
120 FOR C=1 TO 22
130 A=A+0.28: X=SIN(A)*8: Y=COS(A)*8
140 PLOT 4*X+80,4*Y+40
150 NEXT C
```

```
100 REM      ***  ELLIPSE  ***
110 GRAPHICS 7:COLOR 1
120 FOR C=1 TO 22
130 A=A+0.28: X=SIN(A)*4: Y=COS(A)*8
140 PLOT X+80,Y+40
150 NEXT C
```

```
100 REM      ***  BIGGER ELLIPSE  ***
110 GRAPHICS 7:COLOR 1
120 FOR C=1 TO 45
130 A=A+0.28: X=SIN(A)*4: Y=COS(A)*8
140 PLOT 4*X+80,4*Y+40
150 NEXT C
```

Here is an example of the small circle and ellipse:



Notice that the difference in size is controlled by the PLOT commands in line 140 and the size of the FOR-NEXT-loop in line 120. Experiment with these and with the POSITION command. You'll find that you can do very interesting graphic designs with them in ways that almost approximate the ease of Turtle Graphics in Atari Logo and PILOT.

# 36

---

## *The Simple Averager*

This program allows you to average numbers. As you enter a list of numbers, it gives you a running average and a count of how many numbers you have entered. It is convenient for averaging school grades, income, or in the climate we come from, daily rainfall.

```

100 REM *** SIMPLE AVERAGER ***
110 GRAPHICS 0:POKE 752,1
120 ? CHR$(125):POSITION 11,2:? "SIMPL
E AVERAGER"
130 POSITION 2,5:? "ENTER VALUES - PRE
SS START TO CLEAR":POSITION 2,8:? "AVE
RAGE:          VALUES:" :T=0:C=0
140 TRAP 140:POSITION 2,12:? CHR$(156)
:POSITION 2,12:? "?"
150 IF PEEK(53279)=6 THEN 120
160 IF PEEK(764)=255 THEN 150
170 POSITION 2,12:INPUT N
180 C=C+1:T=T+N:POSITION 11,8:? INT(T/
C*100)/100;"      ":POSITION 31,8:? C:GO
TO 140

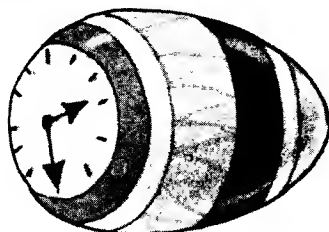
```

# 37

## The Egg Timer

It's not very likely that you have your computer in the kitchen or that you'll use it to time your eggs. However, a timing program is a very convenient insert into other programs or to use to remind you to make a phone call or wake up from a nap. Actually, there are even a few kitchen computer freaks who do use their computers to time their eggs.

In this program you enter the number of minutes or seconds you want to be timed. The computer will display and count down the time, emitting a sound (hopefully pleasant) when it reaches 0. If you want, you can change that sound to a Bach fugue or a Mozart sonata. Just set up a subroutine at line 310 and set to work programming the music of your choice.





```

100 REM      *** THE EGG TIMER ***
110 GRAPHICS 0:POKE 752,1
120 GOSUB 400:POSITION 14,6:? "SET TIM
ER...":POSITION 2,10:? "MINUTES: ";:IN
PUT M
130 POSITION 2,12:? "SECONDS: ";:INPUT
S:IF S<0 OR S>59 THEN POSITION 2,12:P
RINT CHR$(156):GOTO 130
140 GOSUB 400:POSITION 19-LEN(STR$(M))
,7:? M;":":":IF S<10 THEN ? "0";
150 ? S:POSITION 14,10:? "Press START"
160 IF PEEK(53279)<>6 THEN 160
170 POSITION 14,10:? "          ":POK
E 20,0
200 POSITION 18-LEN(STR$(M)),7:PRINT "
":M;":":":IF S<10 THEN PRINT "0";
210 PRINT S
220 IF PEEK(20)<60 THEN 220
230 POKE 20,0:S=S-1:IF S<0 THEN S=59:M
=M-1:IF M<0 THEN 300
240 GOTO 200
300 POKE 764,255
310 FOR P=9 TO 12 STEP 3:FOR T=15 TO 0
STEP -1:SOUND 0,P,10,T:NEXT T:NEXT P
320 C=1-C:IF C=1 THEN POSITION 14,10:P
RINT "PRESS ANY KEY"
330 IF C=0 THEN POSITION 14,10:PRINT C
HR$(156)
340 IF PEEK(53279)=7 AND PEEK(764)=255
THEN 310
350 POKE 764,255:GOTO 120
400 PRINT CHR$(125):POSITION 9,3:PRINT
"*** THE EGG TIMER ***":RETURN

```

# 38

## The Wise Shopper

This program converts the price of a product to its price per ounce or per gram which is useful for comparison shopping. Sometimes, big boxes contain a lot of empty space and buying by unit weight can save you money. To use the program, enter the product's total price

and then choose ounces or grams before you enter the product weight.

```
100 REM *** THE WISE SHOPPER ***
110 GRAPHICS 0:POKE 752,1:OPEN #1,4,0,
   "K:"
120 ? CHR$(125):POSITION 8,2:? "*** TH
   E WISE SHOPPER ***":POSITION 2,6
130 TRAP 120:? "ENTER TOTAL PRODUCT PR
   ICE: ";:INPUT P
140 ? :? "WEIGHT IN (1)OUNCES OR (2)GR
   AMS?"
150 GET #1,K:K=K-48:IF K<1 OR K>2 THEN
   150
160 TRAP 160:POSITION 2,10:? CHR$(156)
   :POSITION 2,10:? "ENTER WEIGHT IN ";:I
   F K=1 THEN ? "OUNCES: ";
170 IF K=2 THEN ? "GRAMS: ";
180 INPUT W:IF K=1 THEN G=INT(28.35*W*
   100)/100
190 IF K=2 THEN G=W:W=INT(G/28.35*100)
   /100
200 POSITION 2,13:? "OUNCES: ";W:POSIT
   ION 2,15:? "PRICE PER OUNCE: ";INT(P/W
   *10000)/100;" CENTS"
210 POSITION 2,17:? "GRAMS: ";G:POSITI
   ON 2,19:? "PRICE PER GRAM: ";:T=INT(P/
   G*10000)/100:IF T>0 THEN ? T;" CENTS"
220 IF T=0 THEN ? "-"
230 POSITION 13,22:? "PRESS ANY KEY":P
   OKE 764,255
240 IF PEEK(764)=255 THEN 240
250 POKE 764,255:GOTO 120
```

# 39

---

## Metric Converts

This is a simple conversion utility. It allows you to convert the following pairs into each other's units of measurement: centimeters/inches; meters/yards; and kilometers/miles.

```

100 REM *** METRIC CONVERTS ***
110 GRAPHICS 0:POKE 752,1:OPEN #1,4,0,
"K:"
120 ? CHR$(125):POSITION 11,2:? "METRI
C CONVERSIONS":POSITION 2,5
130 ? "1. CENTIMETER TO INCH":? "2. ME
TER TO YARD":? "3. KILOMETER TO MILE"
140 ? "4. INCH TO CENTIMETER":? "5. YA
RD TO METER":? "6. MILE TO KILOMETER"
150 GET #1,K:K=K-48:IF K<1 OR K>6 THEN
150
160 POSITION 2,14:FOR T=1 TO 3:? CHR$(
156):NEXT T:POSITION 2,14:ON K GOTO 20
0,210,220,230,240,250
200 ? "CENTIMETERS ";:INPUT N:A=N*0.39
4:? :? "INCHES ";A:GOTO 150
210 ? "METERS ";:INPUT N:A=N*1.094:? :
? "YARDS ";A:GOTO 150
220 ? "KILOMETERS ";:INPUT N:A=N*0.621
4:? :? "MILES ";A:GOTO 150
230 ? "INCHES ";:INPUT N:A=N*2.538:? :
? "CENTIMETERS ";A:GOTO 150
240 ? "YARDS ";:INPUT N:A=N*0.914:? :?
"METERS ";A:GOTO 150
250 ? "MILES ";:INPUT N:A=N*1.6:? :? "
KILOMETERS ";A:GOTO 150

```

Here are the lines where the conversion formulas are stored. You can change them if you want to provide other conversions such as pounds into ounces (using the formula  $\text{OUNCE}=\text{POUND}*16$ ), Fahrenheit into Centigrade, tons into pounds, yards into feet, etc.

```

200 ? "CENTIMETERS ";:INPUT N:A=N*0.39
4:? :? "INCHES ";A:GOTO 150
210 ? "METERS ";:INPUT N:A=N*1.094:? :
? "YARDS ";A:GOTO 150
220 ? "KILOMETERS ";:INPUT N:A=N*0.621
4:? :? "MILES ";A:GOTO 150
230 ? "INCHES ";:INPUT N:A=N*2.538:? :
? "CENTIMETERS ";A:GOTO 150
240 ? "YARDS ";:INPUT N:A=N*0.914:? :?
"METERS ";A:GOTO 150
250 ? "MILES ";:INPUT N:A=N*1.6:? :? "
KILOMETERS ";A:GOTO 150

```

# 40

## Decimal/Hex/Binary Conversion

This program is very useful for people who work with computers. It converts numbers from the binary, decimal, and hexadecimal (base 16) number systems into equivalent form in the other systems. You can start with any of the three systems and get the equivalent in the other two systems. In case you are not familiar with some of these systems of numeration, here is a summary of how they work:

### Base 10: Symbols 0 1 2 3 4 5 6 7 8 9

The base 10 system is the one we all learn first. It has ten symbols and uses the powers of ten to change place value. Thus, counting in base 10 goes as follows:

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17

Note that once all the symbols are used up you move over one place (from 9 to 10). The powers of ten that are given in the following chart determine the values of the symbols:

$10^3$	$10^2$	$10^1$	$10^0$
1000	100	10	1

Thus, the number 1234 actually means:

$$1 \cdot 10^3 + 2 \cdot 10^2 + 3 \cdot 10^1 + 4 \cdot 10^0 = 1234$$

You might remember learning this in fifth or sixth grade though most people usually forget it quickly since they don't need to know it in order to calculate. However, when you move from one base system into another the question of place value is crucial.

## Base 2: Symbols 0 1

There are only two symbols in base 2: 0 and 1, which is why it is so important for computer programming. Every computer is a complex and well-organized collection of ON and OFF switches that can be represented by 1(ON) and 0(OFF). The language your computer understands is base 2 machine language and everything you type in has to be translated into a base 2 system before the computer can process it. Base 2 functions structurally like base ten. It uses place values only they represent powers of 2 instead of powers of 10. Here are the values of the first four places in a base 2 number system:

	$2^3$	$2^2$	$2^1$	$2^0$
value	8	4	2	1

Here are the numbers from 1 to 10 in base 2:

1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010

To understand this, let's take a closer look at 10 which is represented in base 2 as 1010 and translate it to base 10:

$$1010 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 8 + 0 + 2 + 0 = 10$$

Base 2 is a natural language for programming computers. However, programming in this machine language is tedious and leads to many errors. For example, imagine trying to copy this machine language program without making an error or going to sleep:

```
1011100010101010
1100000100011001
0110010101010101
1000001001100010
1111100100010101
```

In order to facilitate programming on a level that is close to machine language and that runs more quickly than BASIC (which has to be translated into machine language internally before it can be RUN by the computer) an intermediate code using base 16 (called the *hexadecimal system*) is often used. The reason for this is that four places in

the binary system (i.e., 2 to the 4th power of  $2*2*2*2$ ) equal 16 and, so, one digit in the hexadecimal system can represent four in the binary system. Here is the hex symbolism. If you want to learn more about this you should consult an introductory book that has a chapter on assembly language programming.

### **Base 16 Number System (Hexadecimal):**

**Symbols 0 1 2 3 4 5 6 7 8 9 A B C D E F**

The letters A, B, C, D, E, F represent the base 10 numbers 10, 11, 12, 13, 14 and 15.

Counting from 1 to 10 in HEX looks like this: 1, 2, 3, 4, 5, 6, 7, 8, 9, A. Here are the HEX place values:

	$16^3$	$16^2$	$16^1$	$16^0$
<b>values</b>	<b>4096</b>	<b>256</b>	<b>16</b>	<b>1</b>

Thus 100A in Hex would be:

$$1*16^3 + 0*16^2 + 0*16^1 + 10*16^0 = 4096 + 0 + 0 + 10 = 5006 \text{ in base 10.}$$

Enough of the details. Here is a conversion program which might come in handy if you do machine or assembly language programming:

```

DECIMAL/HEX/BINARY CONVERSION

DECIMAL: 245
HEX: F5H
BINARY: 11110101B

ENTER VALUE: ?

```

```

100 REM *** DECIMAL/HEX/BINARY ***
110 DIM T$(21),B$(20),H$(5),D$(7),E$(3
7),HH$(16),B(19),H(4):E$(1)=" ":E$(37)
=" ":E$(2)=E$:GRAPHICS 0:POKE 752,1
120 ? CHR$(125):POSITION 5,2:? "DECIMA
L/HEX/BINARY CONVERSION"
130 D$="0":H$=D$:B$=D$:GOSUB 600
140 FOR T=0 TO 4:READ K:H(T)=K:B(T*4)=
K:FOR J=1 TO 3:READ K:B(T*4+J)=K:NEXT
J:NEXT T:HH$="0123456789ABCDEF"
200 TRAP 200:POSITION 2,14:? E$:POSITI
ON 2,14:? "ENTER VALUE: ";:INPUT T$
210 L=LEN(T$):IF T$(L,L)="B" THEN 400
220 IF T$(L,L)="H" THEN 500
230 D=VAL(T$):D$=T$:IF D>1048575 THEN
200
300 R=1:IF D>255 THEN R=3:IF D>65535 T
HEN R=4
310 H$="":K=D:FOR T=R TO 0 STEP -1:J=I
NT(K/H(T)):K=K-J*H(T):H$(LEN(H$)+1)=HH
$(J+1,J+1):NEXT T
350 R=7:IF D>255 THEN R=15:IF D>65535
THEN R=19
360 B$="":K=D:FOR T=R TO 0 STEP -1:J=I
NT(K/B(T)):K=K-J*B(T):B$(LEN(B$)+1)=ST
R$(J):NEXT T:GOSUB 600:GOTO 200
400 D=0:C=0:FOR T=L-1 TO 1 STEP -1:IF
T$(T,T)="1" THEN D=D+B(C):GOTO 420
410 IF T$(T,T)<>"0" THEN 200
420 C=C+1:NEXT T:D$=STR$(D):GOTO 300
500 D=0:C=0:FOR T=L-1 TO 1 STEP -1:A=A
SC(T$(T,T)):IF A>64 AND A<71 THEN A=A-
55:GOTO 530
510 IF A<48 OR A>57 THEN 200
520 A=A-48
530 D=D+A*H(C):C=C+1:NEXT T
540 D$=STR$(D):GOTO 300
600 FOR T=0 TO 2:POSITION 2,5+2*T:? E$
:NEXT T:POSITION 2,5:? "DECIMAL: ";D$:
? :? "      HEX: ";H$;"H"
610 ? :? " BINARY: ";B$;"B":RETURN
800 DATA 1,2,4,8,16,32,64,128,256,512,
1024,2048,4096,8192,16384,32768,65536,
131072,262144,524288

```

# Finale

## 41

---

### A Computer Dating Service

And now, to end the book (almost) on an up note, here is a very liberal dating program. The program is set up to record the answers to 20 questions after taking your name and phone number and assigning you a code number. When you have completed your profile, you can store the program on disk. One person a dating program does not make. Therefore, you have to gather data from a lot of people before it can begin to match their responses and come up with what, from its point of view (really our point of view since a computer does not have convictions or sets up criteria for matching), is your ideal mate from the available pool.





```

100 REM * COMPUTER DATING SERVICE *
110 DIM N$(30),NA$(3000),P$(15),PA$(15
00),T$(40),Q(99,19):OPEN #1,4,0,"K:":N
A$(1)=" ":NA$(3000)=" ":NA$(2)=NA$
120 PA$=NA$:GRAPHICS 0:POKE 752,1
130 GOSUB 900:POSITION 2,6:? "1. Add a
person to dating pool":? :? "2. Selec
t a date"
140 ? :? "3. Save data bank to disk":?
:? "4. Load data bank from disk"
150 GET #1,K:K=K-48:IF K<1 OR K>4 THEN
150
160 ON K GOTO 500,200,800,850
200 GOSUB 900:? "Enter you I.D. number
":? :INPUT N:IF N<0 OR N>=C THEN 130
210 ? :? "FINDING A DATE FOR:":? NA$(N
*30+1,N*30+30):? PA$(N*15+1,N*15+15)
220 DD=0:DT=0:FOR T=0 TO C-1:TT=0
230 IF T=N THEN 400
300 IF Q(N,1)=Q(T,0) THEN TT=TT+50
305 IF Q(T,1)=Q(N,0) THEN TT=TT+50
310 IF Q(N,3)=Q(T,2) THEN TT=TT+25
315 IF Q(T,3)=Q(N,2) THEN TT=TT+25
320 IF Q(N,5)=Q(T,4) THEN TT=TT+25
325 IF Q(T,5)=Q(N,4) THEN TT=TT+25
330 IF Q(N,6)=Q(T,6) THEN TT=TT+50
335 IF Q(N,7)=Q(T,7) THEN TT=TT+50
340 IF Q(N,8)=Q(T,8) THEN TT=TT+50
345 IF Q(N,10)=Q(T,9) THEN TT=TT+25
350 IF Q(T,10)=Q(N,9) THEN TT=TT+25
355 IF Q(N,11)=Q(T,11) THEN TT=TT+25
360 IF Q(N,12)=Q(T,12) THEN TT=TT+50
365 IF Q(N,13)=Q(T,13) THEN TT=TT+25
370 IF Q(N,14)=Q(T,14) THEN TT=TT+25
375 IF Q(N,16)=Q(T,15) THEN TT=TT+25
380 IF Q(T,16)=Q(N,15) THEN TT=TT+25
385 IF Q(N,17)=Q(T,17) THEN TT=TT+25
390 IF Q(N,19)=Q(T,18) THEN TT=TT+25
395 IF Q(T,19)=Q(N,18) THEN TT=TT+25
400 IF TT>DT THEN DT=TT:DD=T
410 NEXT T:? :?
420 ? "Your ideal date is:":? NA$(DD*3
0+1,DD*30+30):? PA$(DD*15+1,DD*15+15):
FOR T=1 TO 1000:NEXT T:GOTO 130
500 GOSUB 900:? "Enter your name":? :I
NPUT N$:NA$(C*30+1,C*30+30)=N$
510 GOSUB 900:? "Phone Number":? :INPU
T P$:PA$(C*15+1,C*15+15)=P$
520 RESTORE 600:FOR T=0 TO 19:GOSUB 90
0:READ T$:? T$:READ T$:? T$
530 GET #1,K:K=K-48:IF K<1 OR K>3 THEN
530

```

```

540 Q(C,T)=K:NEXT T
550 ? :? N$:? "Your I.D. number is ";C
:FOR T=1 TO 1000:NEXT T:C=C+1:GOTO 130
600 DATA Are you (1)Male (2)Female (3)
Both,
605 DATA Do you wish to date a (1)Male
,(2)Female (3)Both
610 DATA Are you (1)Young (2)Middle-ag
ed,(3)Old
615 DATA Is your ideal date (1)Young,(
2)Middle-aged (3)Old
620 DATA How attractive are you (1)Ext
remely,(2)Moderately (3)Not very
625 DATA How attractive is your ideal
date,(1)Extremely (2)Moderately (3)Not
very
630 DATA Where would you like to go on
a first,date (1)Movie (2)Dancing (3)M
otel
635 DATA Are you an (1)Outdoor person,
(2)Indoor person (3)Both
640 DATA How important is sex (1)Very,
(2)Fairly (3)Not very
645 DATA Are you artistic (1)Yes (2)No
,(3)Sort of
650 DATA Would you like your date to b
e,artistic (1)Yes (2)No (3)Sort of
655 DATA How do you feel about kids,(1
)Love them (2)Hate them (3)Both
660 DATA Do you enjoy alcohol or drugs
,(1)Yes (2)No (3)Sometimes
665 DATA Which type of music do you li
ke most,(1)Classical (2)Rock (3)Countr
y
670 DATA How much television do you wa
tch,(1)Quite a bit (2)Some (3)Very lit
tle
675 DATA Are you (1)Wealthy (2)Doing o
kay,(3)Getting by
680 DATA Is your ideal date (1)Wealthy
,(2)Doing okay (3)Getting by
685 DATA Are your habits (1)Unhealthy,
(2)Healthy (3)No habits
690 DATA Are you usually (1)Happy (2)U
nhappy,(3)Both
695 DATA Is your ideal date usually (1
)Happy,(2)Unhappy (3)Both
800 TRAP 820:CLOSE #2:OPEN #2,8,0,"D:D
ATING.DAT":PUT #2,C:FOR T=0 TO C-1:PRI
NT #2;NA$(T*30+1,T*30+30)
810 PRINT #2;PA$(T*15+1,T*15+15):FOR J
=0 TO 19:PUT #2,Q(T,J):NEXT J:NEXT T

```

```

820 CLOSE #2:GOTO 130
850 TRAP 820:CLOSE #2:OPEN #2,4,0,"D:D
ATING.DAT":GET #2,C:FOR T=0 TO C-1:INP
UT #2;T$:NA$(T*30+1,T*30+30)=T$
860 INPUT #2;T$:PA$(T*15+1,T*15+15)=T$
:FOR J=0 TO 19:GET #2,K:Q(T,J)=K:NEXT
J:NEXT T:GOTO 820
900 PRINT CHR$(125):POSITION 9,3:? "CO
MPUTER DATING SERVICE":POSITION 2,6:RE
TURN

```

Notice that the questions are all listed in the following data statements:

```

600 DATA Are you (1)Male (2)Female (3)
Both,
605 DATA Do you wish to date a (1)Male
,(2)Female (3)Both
610 DATA Are you (1)Young (2)Middle-ag
ed,(3)Old
615 DATA Is your ideal date (1)Young,(
2)Middle-aged (3)Old
620 DATA How attractive are you (1)Ext
remely,(2)Moderately (3)Not very
625 DATA How attractive is your ideal
date,(1)Extremely (2)Moderately (3)Not
very
630 DATA Where would you like to go on
a first,date (1)Movie (2)Dancing (3)M
otel
635 DATA Are you an (1)Outdoor person,
(2)Indoor person (3)Both
640 DATA How important is sex (1)Very,
(2)Fairly (3)Not very
645 DATA Are you artistic (1)Yes (2)No
,(3)Sort of
650 DATA Would you like your date to b
e,artistic (1)Yes (2)No (3)Sort of
655 DATA How do you feel about kids,(1
)Love them (2)Hate them (3)Both
660 DATA Do you enjoy alcohol or drugs
,(1)Yes (2)No (3)Sometimes
665 DATA Which type of music do you li
ke most,(1)Classical (2)Rock (3)Countr
y
670 DATA How much television do you wa
tch,(1)Quite a bit (2)Some (3)Very lit
tle
675 DATA Are you (1)Wealthy (2)Doing o
kay,(3)Getting by

```

680 DATA Is your ideal date (1)Wealthy  
, (2)Doing okay (3)Getting by  
685 DATA Are your habits (1)Unhealthy,  
(2)Healthy (3)No habits  
690 DATA Are you usually (1)Happy (2)U  
nhappy, (3)Both  
695 DATA Is your ideal date usually (1  
)Happy, (2)Unhappy (3)Both

The liberality of the program shows in the very first question. It allows for you to be male, female, or both. The second question allows you to date the sexual preferences of your choice. Of course, you might feel this program goes too far. All you have to do is list the data statements and change them to suit your own ways of looking at the problem of finding a best mate or date. Just be sure that each change you make goes on one data line. The weighting factors built into the program can also be changed if you have some programming experience. You'll find them in the 300's.

## 1/2 of Many Things You Can Do With Your Computer

You can use the programs as they are listed in this book. You can also consider them program forms that provide a framework for developing new programs on your own. Here are a number of ideas that take off from the 41 programs we have worked out. They provide you 1/2 of what you need to develop applications for your computers. The other half, the new programs themselves, are for you to work out.

### **1 An All-Purpose Instrument Tuner**

One of the programs in this book provides a guitar tuner. Using the sound capabilities of your computer, you can also create a saxophone, clarinet, violin, fiddle, trumpet, or all-purpose instrument tuner. Instead of string diagrams, you would have to suit the graphics to the

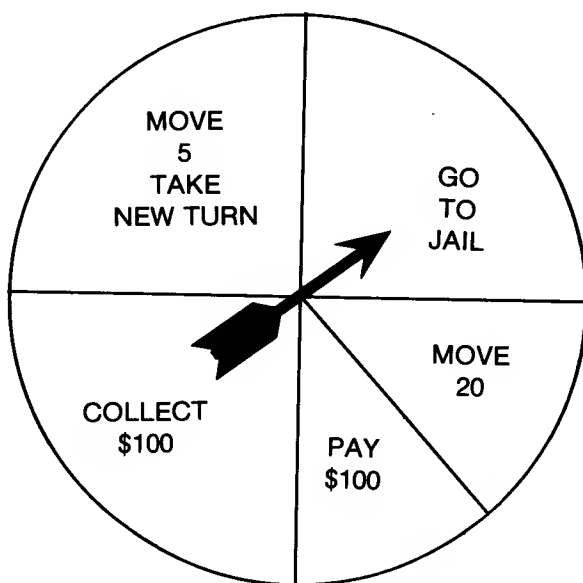
nature of the different instruments and, therefore, have many different graphics screens in your program.

## 2 A Music Tutor

This program would extend the notion of a guitar chord tutor. It could show the fingering of woodwinds, the valve positions on brass, etc. You could even develop an extensive music teaching program.

## 3 A Game Spinner

You can use some of the same ideas used in the Dungeons and Dragons dice maker to create a game spinner. The spinner differs from a usual die in that it can have the same values on different sections, or it can have verbal instructions or game moves on it. Different sections can also be weighted so that there is not the same probability of getting each possible result. Here is an example of what a spinner might look like:



On the sample spinner there is a greater chance of landing in jail than collecting money. Notice that the instructions on the spinner refer to game moves and to paying or collecting money. These different modes and weights can easily be incorporated into a customized game spinner program.

#### **4 A Split Screen Note Pad**

Try to create a note pad that splits the screen in half so that you can write on either half, or write on one half and make sketches and notes on the other half. Try to build in a way to press a letter or number which allows you to control which part of the screen you use.

#### **5 A Calculator with Memory**

Add a memory storage capability to the calculator program so that the results of any calculation can be stored and then used at another time.

#### **6 Animated Guessing Games**

Modify the Gestalt guessing game so that you see small figures running all around the screen and have to guess how many there are. Remember that animation can easily be done using Control graphics and clearing the screen after each positioning of your figures. When you draw a figure, clear the screen and then draw the figure in a slightly different position or posture you create the illusion of animation.

#### **7 Mandala Meditation**

Create your own meditation programs playing with the color function of your computer. Add sound to the program and see if you can create a multimedia work of art.

## **8 Pie Chart and Line Graph Generator**

The bar graph generator presented in the book is only one of many graph forms that can be programmed. Try to make a line graph. (Remember that the bar graph can easily be turned into a line graph by turning it on its side and drawing lines from the top points.) Also create a pie graph generator. A pie graph generating program can also be used in a game spinner generating program.

## **9 Multiple Sorting Address Card File**

Modify the address file program so that you can sort by state, city, phone number, and comments as well as by name. What this requires is that you build routines similar to the one used in the program in this book to do name sorting for each line in the address book entry.

## **10 Profile Mug Shots**

Using character graphics, change the Mug Shot program from one that uses full faces to one that uses profiles.

## **11 Multilingual Dictionary**

Create a multilingual dictionary. For example, create a dictionary in five languages. Set up the program so that you have a few entries to begin with (for example all the pronouns in five different languages) and then new entries can be added and saved on disk. This way you will not merely have a multilingual dictionary but a dictionary form. As you learn more vocabulary, you build your dictionary. Many of the routines needed to create such a program can be taken from programs listed in this book (the disk storage routine, for example).

## **12 Prime Number Generator**

Create a program that starts at 1 and lists the first  $n$  prime numbers for any integer  $n$ . Add another routine to the program in which you

input any integer and the computer tells you whether it is prime or not.

### 13 Least Common Denominator

Here is another homework helper program. Set up the computer so that it will list all the common denominators of any two or three numbers and also list the least common denominator.

### 14 A Book Title Generator

We included an excuse generator in our 41 programs. See if you can create a book title generator using some of the common book title forms such as:

A ..... (1) ..... in the ..... (2) .....

List (1)	List (2)
flash	pan
day	life
scream	wilderness

giving such variations as

*A Flash in the Wilderness* and

*A Day in the Pan.*

Some other title forms to play with are: The ..... (1) of the ..... (2) ..... and To ..... (1) ..... a ..... (2) .....

### 15 A Code Maker

Create a program that will scramble the whole alphabet (instead of just a word as in our Word Scrambler) and then turn that scrambling into a code. The program should let you put in a word or sentence and give you the coded version. It should also let you put in a message in that code and translate it back into English.



\$6.95

## 41½ FUN PROJECTS FOR THE ATARI®

Dale Disharoon & Herbert Kohl

Do you remember the day you brought your shiny new Atari home? What an exciting time! At first it seemed as if you could think of thousands of ways it would make your life easier, better, and more fun. However, if you're like most people, as the months rolled by and the novelty of owning a computer subsided, routine chores slowly replaced that initial anticipation.

**41½ FUN PROJECTS FOR THE ATARI®** gives you back that wonderful excitement. One of the first Creative Pastimes books, from the Reston Computer Group, this fantastic book shows you 41½ good reasons to fall in love with your Atari all over again.

In a burst of creative energy, Dale Disharoon and Herbert Kohl deliver some fresh ideas that you'll find easy to key in and modify. The 41½ projects range all the way from a dating program to a mini word processor. You'll be absolutely obsessed with a Dungeons and Dragons dice rolling program, and you'll probably want to take full advantage of the innovative strategy and music games. Practically speaking, programs such as a guitar tuner and chord teacher, a metric converter, and a comparison shopping program truly make your life easier.

**A Creative Pastimes Book**  
**RESTON PUBLISHING COMPANY, INC.**

*A Prentice-Hall Company*  
Reston, Virginia



21898 21015

